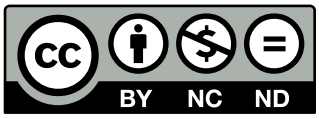
Universidad Carlos III de Madrid

Trabajo de Fin de Grado
Grado en Ingeniería de Sistemas de Comunicaciones

Estudio de movilidad para tráfico IP Multicast en un emulador de redes

Autor: Alberto del Río Ponce
Tutor: Ignacio Soto Campos
Fecha: 20 de febrero de 2018



Resumen

Son las nuevas tendencias con una aparición masiva de nodos móviles en la red, las que crean un interés por buscar soluciones en el ámbito de la movilidad, siendo el objetivo del proyecto el estudio del intercambio de tráfico multicast en nodos móviles. Sin usar una solución específica, se extraen ideas de una solución de movilidad (PMIP) para la consecución del objetivo.

A lo largo de la memoria se detallan las diversas herramientas utilizadas, partiendo de un emulador de redes denominado CORE y las implementaciones de protocolos que permitan transmitir tráfico. Para la consecución del tráfico, se estudian protocolos de routing unicast y multicast. En el caso unicast, Routing Information Protocol (RIP) y Open Shortest Path First (OSPF) y sus versiones adaptadas a direcciones IPv6, para determinar las rutas entre nodos. Para multicast, se estudia un protocolo de routing multicast, Protocol Independent Multicast (PIM) que se encarga de que los routers entiendan como reenviar el tráfico de las fuentes a los nodos suscritos a los grupos multicast. Los protocolos de routing se consiguen obtener gracias a herramientas que los implementan, como Pimb, XORP y MRD6.

Finalmente, se estudia la recepción de tráfico multicast en movilidad y se obtienen unos malos resultados. Según las pruebas obtenidas, se obtienen retrasos de hasta un minuto en la recepción de tráfico en movilidad debido a la frecuencia de envío de mensajes MLD. Se plantan las bases de futuros estudios, entre ellos, una posible mejora en la transmisión de los mensajes MLD para mejorar las prestaciones. Adicionalmente, se estudian posibles vías de expansión de mercado, siendo esta memoria únicamente una herramienta para el estudio de movilidad en tráfico multicast, sin ánimo de lucro a corto plazo.

Palabras clave: Mobile radio mobility management, IP networks, Internet, IP Multicast, Mobility

Abstract

The proliferation of scenarios with massive numbers of mobile nodes in the network creates a research opportunity due to the necessity of solutions in the field of mobility. In this way, the aim of this project is to tackle this issue carrying out a study of the multicast traffic exchanged in mobile nodes. Without using a specific solution, we extract information from a mobility solution (PMIP) for the achievement of the objective.

Throughout the memory are detailed the various tools with which the project is developed, based on a network emulator called CORE and the protocols implementations that allow traffic to be transmitted. For the data transmission, it is necessary to study the unicast routing protocols. In this case, both Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) and their versions adapted to IPv6 addresses have been thoroughly studied. In order to get multicast traffic, a multicast routing protocol, Protocol Independent Multicast (PIM), is studied, which allows nodes to subscribe to multicast groups and receive traffic of this type. The routing protocols are obtained thanks to tools that have them implemented internally, such as Pimb, XORP and MRD6.

Finally, we check the functionality of multicast traffic reception in mobility, and the results obtained are bad. According to the test, delays transmissions in mobility are up to one minute due to the sending frequency of MLD messages. We show future studies and research, among them, a possible improvement in MLD messages transmissions. Finally, we study possible ways of market expansion. However, we emphasize the fact that this thesis is a mere tool for mobility and multicast traffic, with no short-term profit motivation.

Keywords: Mobile radio mobility management, IP networks, Internet, IP Multicast, Mobility

Agradecimientos

Esto va dedicado a todas aquellas personas que han estado a mi lado durante todos estos años, que con su ayuda y apoyo han hecho posible el finalizar exitosamente una de mis etapas de la vida.

Principalmente agradecer a mis padres, que han estado desde el minuto uno que tomé aliento en esta vida. Aunque no suela decirlo con palabras habitualmente, agradezco cada minuto de sacrificio de sus vidas que han facilitado que el día de mañana pueda disponer de un título de ingeniero. No es únicamente el apoyo durante esta etapa, si no la educación que ellos me han instruido desde el primer momento.

Al resto de mi familia, por toda esa ilusión por continuar adelante y felicidad que habéis compartido conmigo.

A todos mis amigos y compañeros de Universidad, por esos momentos de dudas y bajos ánimos que ellos convirtieron en ganas y motivación. Por ser las personas con las que más tiempo he compartido y más he crecido como persona.

A mi tutor, Ignacio, por todos los conocimientos que me ha transmitido durante estos meses, así como por su paciencia y ayuda constante.

Por esas personas que hoy no están, pero también fueron partícipes de lo que soy hoy en día.

A todos vosotros, muchas gracias.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Con la aparición de nuevos dispositivos, principalmente el Smartphone, el vídeo por internet ha aumentado su volumen exponencialmente [1], y con vistas a que esta tendencia siga creciendo. Debido a esto, el tráfico de IPTV (en general, envío de canales de TV tradicionales, o emisiones de eventos en directo) ha cobrado una importancia elevada en internet. La forma más eficiente de transmitir IPTV en la red es de manera multicast, método que usan las propias operadoras (Telefónica, Vodafone...) para transmitir sus propios canales de televisión, permitiendo un uso más eficiente del ancho de banda en sus redes en comparación al envío unicast.

Tanto los servicios de IPTV como las transmisiones en directo en plataformas como YouTube son en gran parte consumidos desde dispositivos móviles. Esto se interpreta como un cambio en la forma de consumir contenido, con unas necesidades de poder acceder en todo momento a estos servicios. Esto implica la movilidad de los dispositivos mientras se recibe tráfico multicast.

A simple vista con la expresión *dispositivo móvil* pueda llegarse a pensar en esto como únicamente el Smartphone, pero con dirigirse a cualquier blog o feria tecnológica puedes darte cuenta de las nuevas aplicaciones y dispositivos que nos rodean. Todo está llevando a un mundo de todo conectado, y un ejemplo puede ser el sector del automóvil [2], con los sistemas autónomos de conducción. Se necesita de un equipamiento y sistemas capacitados para que esto sea posible.

1.1. Motivación

Finalmente estas novedades acabarán siendo algo parte habitual de nuestras vidas, por lo que se antoja importante el continuar las investigaciones. De acuerdo a esto, en este trabajo se ha realizado el estudio de tráfico multicast en situaciones de movilidad, destacando el uso de direcciones IPv6 para los nodos.

Para entender mejor el trabajo conviene definir con precisión las diferentes alternativas que existen para el envío de tráfico en Internet [3]:

- Unicast. Se envía individualmente el contenido a un único recipiente, propio de servicios de comunicaciones personales.
- Broadcast. El contenido es enviado de manera masiva, a todos los destinos, véase la televisión o la radio tradicional (es decir, contenido no transmitido por Internet).
- Multicast. Únicamente aquellos receptores suscritos al grupo multicast recibirán el contenido, con ejemplos como IPTV.

IPTV puede ser enviado por unicast, pero conlleva enviar un flujo específico al destinatario consumiendo un ancho de banda superior. Como curiosidad, dentro de los servicios propios de televisión de los operadores, para poder rebobinar programas de televisión al comienzo de éste, es obligado el transmitir el contenido de manera unicast a este usuario. Es decir, en el peor caso un flujo multicast se convierte en unicast (tantos como usuarios que han usado la función de rebobinar).

1.2. Objetivos

El objetivo principal de este proyecto es el estudio del intercambio de tráfico multicast en nodos móviles, es decir, en nodos que cambian su punto de conexión a la red mientras se comunican.

Se organiza mediante los siguientes objetivos parciales:

- Estudio del funcionamiento del tráfico multicast IP.

- Búsqueda de herramientas para la evaluación de intercambio de tráfico entre nodos.
- Estudio de soluciones de soporte de tráfico multicast en movilidad.
- Generación de tráfico multicast hacia nodos receptores suscritos al grupo multicast.
- Evaluación de prestaciones sobre los efectos generados debido a la movilidad con tráfico multicast en diversos escenarios de red.

1.3. Metodología

Para llegar al objetivo de estudio final requerirá pasar por ciertas fases que concluirán con los resultados finales.

- Fase de investigación, en la que se busca información acerca de las tecnologías aplicadas y relacionadas para conocer mejor el ámbito de trabajo.
- Fase de análisis, en la que se trata el alcance del proyecto así como concretar los diferentes protocolos de routing.
- Fase de desarrollo, con la cimentación de las ideas del proyecto, generación del código necesario y aplicación a los entornos para el proyecto.
- Fase de pruebas o validación, en la cuál finalmente se realizarán los test pertinentes para comprobar la viabilidad del tema propuesto y poder sacar conclusiones.

1.4. Organización de la memoria

El presente documento sigue la siguiente estructura:

- Capítulo 2. Se describe el estado del arte referente a todos los conceptos utilizados a lo largo de la memoria. Concretamente, son los diferentes protocolos utilizados, tanto de routing unicast como multicast así como las técnicas y

componentes que hacen posible el funcionamiento de dichos protocolos. Adicionalmente, se explican los frameworks que se estudian durante el proyecto que ponen a disposición los diferentes protocolos de routing.

- Capítulo 3. Se abordan en el marco regulador los diferentes estándares que aplican como medida de investigación al proyecto.
- Capítulo 4. Se comienza a aplicar el envío de tráfico, con el progreso desde tráfico unicast hasta la consecución de tráfico multicast.
- Capítulo 5. Tras la consecución de tráfico multicast, se crean escenarios formados por redes inalámbricas para permitir la movilidad entre los nodos, extrayendo resultados de las diferentes pruebas realizadas.
- Capítulo 6. Gracias a los resultados obtenidos, se extraen conclusiones para abordar futuras líneas de proyectos.
- Capítulo 7. Resumen en inglés de 5 páginas que aborda todo el concepto del proyecto explicando de manera simplificada la memoria.
- Apéndices incluidos en el proyecto, que forman parte de la memoria pero no incluidos en el cuerpo ya que no son fundamentales para la comprensión del proyecto. Entre ellos se encuentra un programa desarrollado para el envío de tráfico; una herramienta que se estudia durante el cuerpo para el routing multicast, pero con un protocolo diferente al del propio cuerpo; el entorno socioeconómico en el que se habla del potencial económico y social del multicast y de la movilidad, con diversas aplicaciones con gran potencial; y finalmente, el presupuesto que origina el proyecto.
- Lista de acrónimos y glosario, que disponen de palabras o expresiones a conocer para la comprensión correcta de la memoria.
- Bibliografía, con las referencias a la documentación utilizada para poder completar las diferentes secciones de la memoria.

Capítulo 2

Estado del Arte

En este capítulo vamos a revisar los conceptos teóricos relacionados, con el objetivo de poder comprender con facilidad el posterior desarrollo del trabajo.

2.1. Conceptos Generales

Las ventajas de transmitir contenido de manera multicast en comparación a unicast son visibles en el ancho de banda consumido, asumiendo por tanto unos menores costes y una escalabilidad controlada. El Internet Engineering Task Force (IETF) ha estandarizado varias soluciones de movilidades pero pensadas principalmente para el tráfico unicast, por lo que se describen dos de las soluciones más populares y qué sucede con ellas en el tráfico multicast.

2.1.1. Mobile IP

Mobile IP (MIP) [4] surge ante el problema de que IPv4 se desarrolló pensando que la dirección IP debía ser tanto una manera de identificar a los dispositivos como de ubicarlos en la red. Al entrar en temas de comunicaciones y movilidad, este concepto origina problemas, ya que cuando un nodo cambia de punto de unión a la red, cambiará su dirección IP y, por tanto, su identificación, con lo que se perderán los paquetes enviados a dicho nodo.

Con MIP [5] se ofrece solución para solventar el problema, ya que asigna a los nodos dos direcciones, *home address* y *care of address*. Una dirección permanente para identificar a un nodo dentro de todo Internet, y otra para localizar la posición, respectivamente. La *home address* actúa como la manera clásica de las direcciones IP, manteniéndose fija a pesar de la movilidad en la red, mientras que la *care of address* variará según cambie su punto de unión a la red.

Dentro de la red es el Home Agent (HA) quien se encarga de asociar ambas direcciones. Cuando el Mobile Node (MN) cambia de punto de conexión a la red debe mandar mensajes de actualización para notificar al HA que ha cambiado su *care of address*. En el momento de transmitir tráfico, se necesita que los niveles por encima de IP sigan viendo como dirección destino (del MN) la *home address*. Se crea un túnel con cabecera IP externa, con dirección destino la *care of address* para encaminar el paquete al punto en el que está el nodo móvil, y con cabecera IP interna con dirección destino la *home address*, de modo que en el destino se procesa la cabecera externa, quedando únicamente la interna. Esta cabecera interna se vuelve a procesar consiguiendo que los niveles superiores sigan viendo como dirección destino la *home address*.

MIP se diseñó para tráfico unicast, pero el tráfico multicast tiene un encaje ahí (aunque no se haya pensado en una solución para ser eficiente con el tráfico multicast). El MN tiene dos opciones:

- Usar el HA como siguiente salto del tráfico multicast. El MN envía mensajes al HA, y éste reenvía el tráfico multicast por el túnel (cabecera externa, dirección destino=*care of address*; cabecera interna, dirección destino=dirección multicast).
- La otra alternativa es que el MN usa la *care of address* para registrarse a los grupos multicast en cada subred que visita (es decir, no usa MIP para el tráfico multicast).

2.1.2. Proxy Mobile IP

Proxy Mobile IP (PMIP) [6] extiende las funcionalidades de MIP, pero con matices que completan el objetivo de la movilidad. Una de las principales diferencias

es que en PMIP es la propia red quién se encargará de rastrear todos los cambios de movilidad en vez del propio MN.

Se introducen una serie de elementos dentro de la arquitectura que puedan manejar la movilidad de manera eficiente sin la participación directa del MN. Principalmente, dentro de la arquitectura destacan el Local Mobility Anchor (LMA) y el Mobile Access Gateway (MAG), encargados de gran parte de la gestión de PMIP. El Dominio Proxy Mobile IP (Dominio PMIP) es el alcance de influencia del protocolo y dentro de éste puede haber varios LMA y MAG para un grupo de MNs.

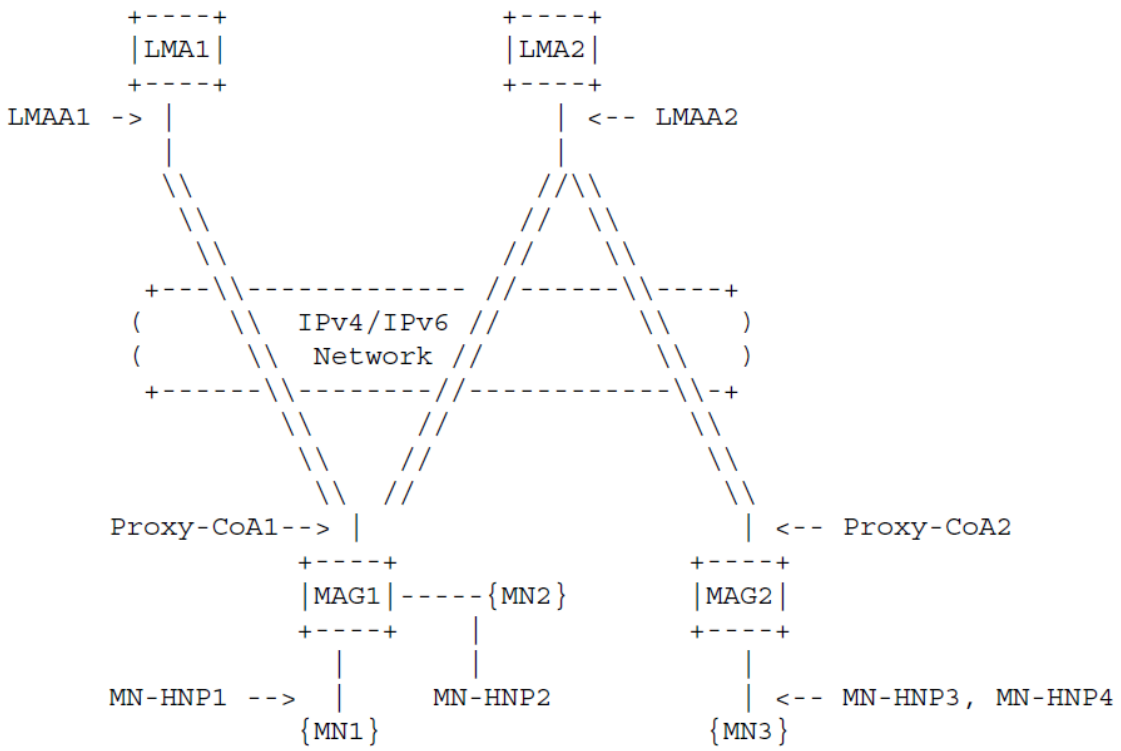


Figura 2.1: Mobile IPv6 Domain (extraído de [6])

Al haber liberado al MN de las funciones de notificar la movilidad [7], otro elemento debe ser quién lleve a cabo estas tareas. En este caso, será el MAG el que vaya registrando todos los movimientos del MN dentro del Dominio PMIP. Una vez que el MAG detecta un nuevo MN en la red deberá comprobar que reúne los requisitos de movilidad para poder ser gestionado, mandando un mensaje Proxy Binding Update (PBU) al LMA, para solicitar la unión del nodo a la red y notificando la localización actual del nuevo nodo.

En el caso de que el MN sea validado, el LMA devolverá al MAG un mensaje Proxy Binding Acknowledgement (PBA) para añadirle dentro del Dominio PMIP, asignando al MN un Mobile Node's Home Network Prefix (MN-HNP), un prefijo para poder ser identificado dentro del dominio. Finalmente, se crea un túnel bidireccional entre el MAG y el LMA.

Los paquetes enviados por el MN serán recibidos por el MAG y, posteriormente, transmitidos por el túnel bidireccional, por lo que puede decirse que el MAG siempre actuará como el router por defecto en el enlace punto a punto con el MN. En el caso de que el destino se encuentre dentro del Dominio PMIP, el MAG puede retransmitir directamente los paquetes al destino sin necesidad de enviarlo a través del túnel.

Los paquetes recibidos de manera externa al Dominio PMIP, son recibidos por el LMA, y éste se encarga de reenviarlos por el túnel bidireccional hacia el MAG. A continuación el MAG elimina el encapsulado y retransmite el paquete por el enlace hacia el MN.

Cuando el MN cambia su punto de unión con la red, el MAG lo detecta y avisa al LMA para que elimine el enlace entre ambos y borre las rutas hacia dicho MN de su tabla de enrutamiento. Una vez se ha conseguido que el MN establezca un nuevo punto de enlace con el MAG, señalizará al LMA con un nuevo PBU para completar la señalización. Resaltar que mientras que el MN permanezca dentro del Dominio PMIP, mantendrá su dirección IP.

Para la gestión del tráfico multicast, hay dos posibles conceptos en PMIP, con diferencias en la arquitectura:

- Un MAG es conectado a diferentes LMA, recibiendo el mismo tráfico por diferentes caminos, dependiendo de las suscripciones de los MN a los diferentes LMA. Esto originaría una acumulación de tráfico redundante en los MAG.
- Cada MAG se configura para estar conectado a un único LMA para todos los MN. En este caso, la redundancia de tráfico la tendría el LMA provocando una avalancha de transmisiones.

2.1.3. Protocolo de comunicaciones IPv6

IPv6 es la versión actualizada del protocolo de comunicaciones de internet. Surge como extensión y como intento de reemplazo de IPv4, motivado en parte por el incremento exponencial de nodos conectados a la red.

Este protocolo dispone de ciertas novedades como la asignación de nuevos rangos de direcciones o mensajes de anuncio automáticos sobre cambios de nodos en la red. Además, se implementa en la propia especificación de IPv6 la transmisión multicast, siendo algo opcional en IPv4. Gracias a esta mejora unido al interés de transmisiones multicast, justifica el uso de direcciones IPv6 durante el proyecto.

2.2. Protocolos Routing

Dado que más adelante es necesaria la configuración de ciertos protocolos en los routers del escenario, hay que estudiar todos aquellos protocolos que se utilizarán para tener una idea del comportamiento de cada uno.

2.2.1. RIP

Routing Information Protocol (RIP) [8] es un protocolo de routing unicast que se encuentra dentro de los llamados Interior Gateway Protocol (IGP). Una vez activado el protocolo y en funcionamiento, cada router anuncia periódicamente (cada 30 segundos) su tabla de enrutamiento sobre todas sus interfaces. Cada router actualizará su propia tabla de enrutamiento comparándola con las que reciba de otros routers.

La forma de controlar que todos los routers del escenario se encuentren activos, es que, si un router deja de responder a los anuncios, se parará de enviar tráfico por las rutas que tengan como destino dicho nodo. A pesar de esto, el router destino que deja de responder a los anuncios sigue presente dentro de la tabla de routing del resto de routers, con la salvedad de que se le asigna la etiqueta de destino inalcanzable. Esto es así para que los routers sepan cómo interpretar el tráfico con destino a dicho nodo, evitando las demoras de tener que volver a establecer el destino como inalcanzable.

Para la gestión de rutas, RIP hace uso de una métrica propia. Es decir, la distancia entre dos elementos la contabiliza mediante el número de saltos que se producen desde el nodo inicial al nodo destino. Se cuenta como un salto cada vez que el tráfico llega a un nuevo router o host de la red. Esta manera de distancia sirve al protocolo para elegir entre dos rutas para un mismo destino. El límite máximo de saltos estandarizado dentro de RIP es de 16, punto a partir del cuál toda red será considerada inalcanzable.

RIPng extiende la funcionalidad de las primeras versiones de RIP haciendo compatible este protocolo con el uso de direcciones IPv6 [9].

2.2.2. OSPF

Open Shortest Path First (OSPF) [10] es otro protocolo routing unicast que se encuentra encuadrado igualmente dentro del grupo de IGP. OSPF ha sustituido a RIP en redes muy grandes, como son las redes empresariales. Al contrario que RIP que transmite su tabla de enrutamiento cada 30 segundos a sus nodos vecinos, OSPF envía esos mensajes de manera multicast a los demás nodos de la red cuando detecta cambios. Adicionalmente, OSPF únicamente manda los cambios que se hayan producido en en los enlaces locales del router, no la información de la tabla al completo.

Con el protocolo RIP se cuenta con una métrica específica para la decisión de rutas según el número de saltos para llegar a un destino. En OSPF esta métrica varía teniendo en cuenta otras variables de la red para cada enlace, pudiendo un operador asignar diferentes costes dependiendo de la situación de los enlaces. Por ejemplo, para un enlace Wireless LAN (WLAN) el coste puede ser inferior que para un radioenlace.

En el proyecto se utiliza OSPFv3, que añade compatibilidad con direcciones IPv6 ya que OSPFv2 únicamente tiene soporte para direcciones IPv4 [11].

2.2.3. PIM

Uno de los protocolos más habituales cuando se habla de routing multicast es el llamado Protocol Independent Multicast (PIM). Esto es gracias a que dispone

de ciertas ventajas dentro de su arquitectura, como el mantener el modelo tradicional de unión a grupo multicast iniciado por el receptor e incluso soporta tanto árboles compartidos como árboles de distribución de camino más corto. Sus dos posibles variantes son PIM-Sparse Mode (PIM-SM) y PIM-Dense Mode (PIM-DM).

El primero de ellos, PIM-SM se basa en la idea de que dentro de la propia red, la mayoría de subredes no requerirán ninguno de los paquetes multicast que se envíen, de ahí su nombre, *Sparse Mode*. Son los propios receptores los que mandan mensajes explícitos sobre el interés de recibir paquetes multicast, formando junto al resto de routers y receptores, árboles de distribución compartida. Como se ve en la figura 2.2, el punto de convergencia de dichos árboles compartidos es el RendezVous Point (RP), el cuál será una dirección IP de un router concreto, al que se le manda de manera unicast la información encapsulada.

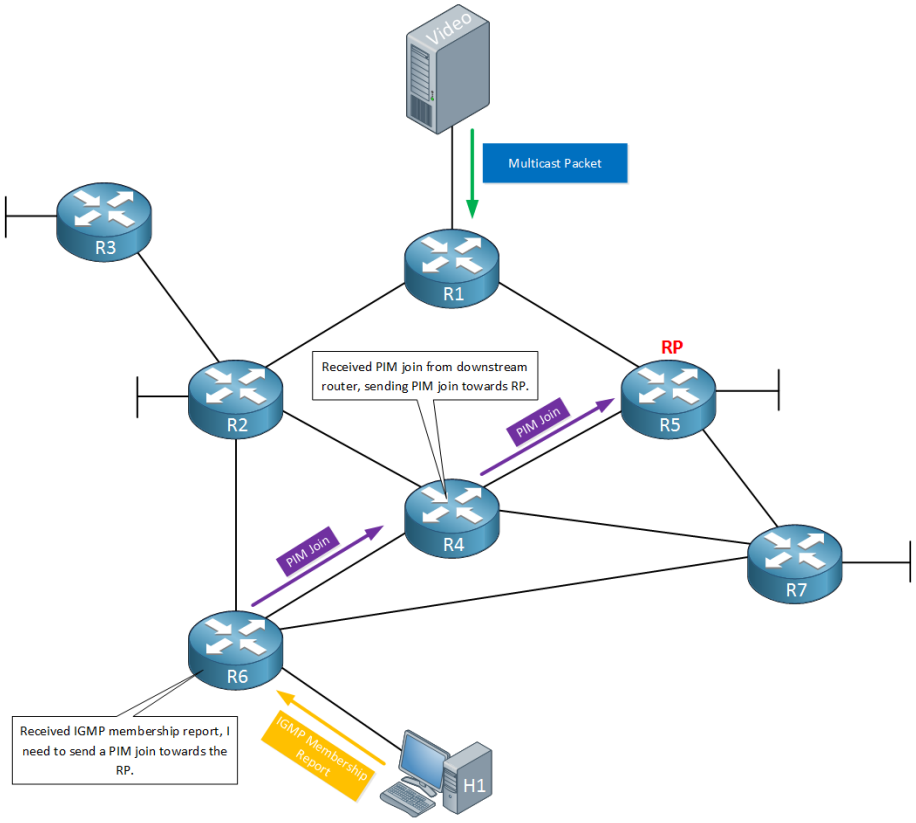


Figura 2.2: PIM-Sparse Mode (extraída de [12]).

Para designar el RP y su posterior descubrimiento por parte de la red, existen varios métodos, pero centrando su atención concretamente en dos. Primero, mediante configuración estática asignando una dirección IP fija al RP o, con el

segundo método, mediante la selección del Bootstrap Router (BR), que asigna el RP de manera automática gracias a la propia red.

Dentro del método bootstrap se diferencian los Candidate Bootstrap Router (Candidate BSR), que coleccionan información sobre el resto de RPs disponibles; y, por otro lado, los Candidate RendezVous Point (Candidate RP) que serán quienes muestren a la red la intención de convertirse en RP. Cada Candidate RP se configura con un prefijo de direcciones multicast que especifica el rango de direcciones multicast que van a poder suscribirse al grupo multicast. Entre los routers que formen la subred se decide quién será el RP mediante el intercambio de mensajes. Finalmente se fijará mediante una serie de criterios, empezando por el *longest match* según la dirección multicast a la que estemos transmitiendo, siguiendo por la prioridad de cada router, que será designada en cada router en su configuración inicial, y, para terminar, con una comparación entre el mayor valor según una función hash. En el caso de que con estos criterios no se haya podido decidir un RP, el que tenga un número mayor de dirección IP será el elegido.

El proceso que sigue desde que existe una fuente multicast y se sirve el tráfico multicast a un nuevo nodo es el siguiente:

- Una fuente puntual está transmitiendo tráfico multicast a un grupo de nodos.
- Los routers mandan periódicamente mensajes Multicast Listener Discovery (MLD) para descubrir nuevos nodos interesados en unirse al grupo multicast.
- Un nuevo MN quiere recibir tráfico multicast. Este MN debe unirse al grupo multicast y para ello debe responder con un mensaje MLD para reportar su intención de recepción de tráfico multicast.
- Se incluye este nuevo nodo dentro del grupo multicast.
- Con las tablas de routing unicast existentes, PIM utiliza Reverse Path Forwarding (RPF) para comprobar las rutas hasta el nuevo destino y evitar bucles en la red.
- El MN empieza a recibir tráfico multicast.

La segunda de las variantes es PIM-DM, que es prácticamente justo lo contrario que PIM-SM, en el que esta vez se asume que la mayoría de receptores

van a solicitar la recepción de paquetes multicast y todos ellos están repartidos de manera bastante densa dentro de la red. PIM-DM no utiliza RPs, ya que utiliza árboles basados en fuente, por lo que se comporta de manera más eficiente cuando la mayoría de receptores están interesados en los datos multicast a recibir, pero es poco escalable ante un dominio con alto número de receptores que no quieren recibir dicho tráfico.

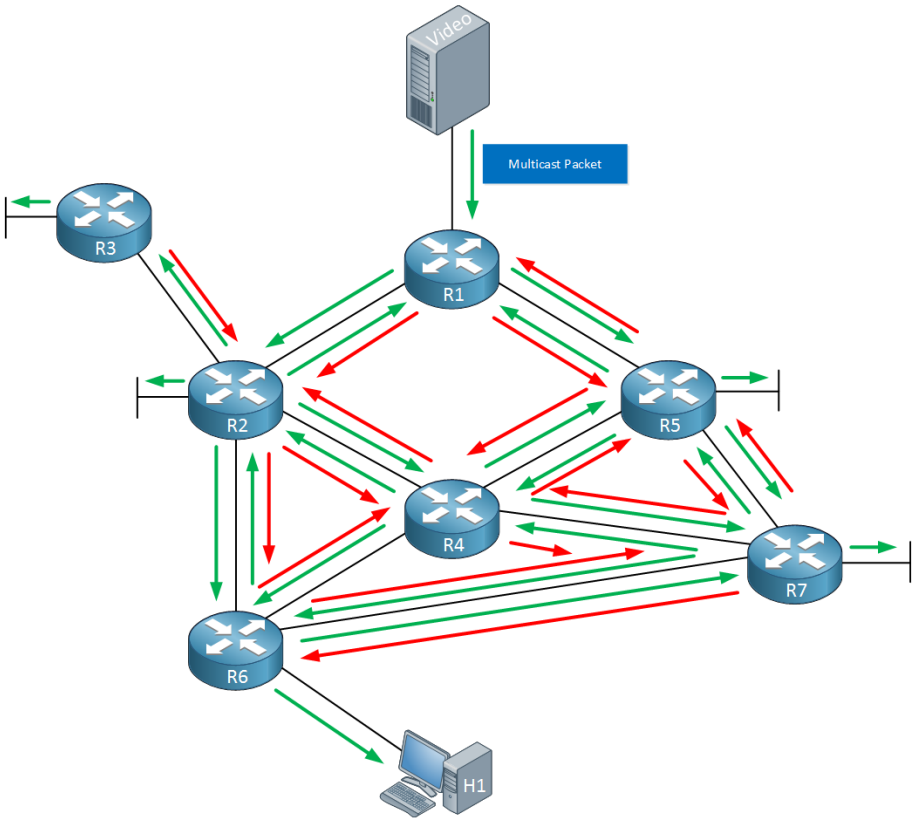


Figura 2.3: PIM-Dense Mode (extraída de [13]).

Su funcionamiento empieza con el transmisor enviando los datos a la red, y cada router lo reenvía a sus vecinos dentro de su área local hasta que en una de las interfaces de transmisión *downstream* se solicita la finalización de reenvío de datos. Esto es debido a que el sistema se basa en inundar todas las interfaces para que llegue a todos los destinos, con la posible generación de bucles en la red. Se puede ver visualmente en la figura 2.3.

2.3. Componentes y Técnicas

2.3.1. MLD

Multicast Listener Discovery (MLD) [14] forma parte de todos los protocolos relacionados con IPv6, y se encuentra embebido en ICMPv6. Con MLD, los routers de una subred descubren los sistemas interesados en suscribirse a un grupo multicast, por lo que en conjunto con un protocolo de routing multicast como PIM, reciben el tráfico multicast y lo retransmiten al nodo destino. Se envían periódicamente mensajes MLD para descubrir nuevos nodos interesados, actualizando los grupos multicast.

Se distinguen dos implementaciones MLD. Por un lado tenemos las funciones propias de cliente, integradas en Linux (o los diferentes Sistemas Operativos), para propósito general. Por otro lado, las implementaciones en servidor, que serán los mensajes MLD que transmitirán nuestros routers al activar protocolos de routing en nuestros escenarios. En ésta última nos referimos a las implementaciones que forman dentro del software adicional que instalamos para habilitar protocolos de routing.

2.3.2. RPF

En conjunción con los protocolos de routing multicast (ejemplos como PIM), se utiliza el llamado Reverse Path Forwarding (RPF) [15]. Esta técnica se utiliza para evitar bucles de reenvíos en la red.

Estos bucles se originan ya que, en principio sin usar esta técnica, un paquete de tráfico multicast puede llegar a un router por dos interfaces diferentes. Esto ocasiona que un paquete que llega por una interfaz determinada, sea reenviado por otra interfaz por la que justamente se está recibiendo el mismo paquete.

Como los protocolos de routing multicast se basan en estructurar la red en árboles compartidos, la ruta del tráfico multicast seguirá el mismo camino pero de manera inversa al receptor. Es decir, un router acepta un paquete desde una fuente a través de una interfaz, únicamente si esa interfaz es la que usaría el router para reenvíar tráfico para llegar a dicha fuente.

2.4. Frameworks

El estudio del trabajo se debe realizar en un escenario lo más cercano a la realidad, por lo que un despliegue de equipamiento de red real puede pensarse como la manera más viable de efectuarlo, consiguiendo unas pruebas finales totalmente fidedignas. El problema de este caso es que sería una situación costosa por toda la diversidad de equipamiento, severamente complejo e inflexible debido a la dificultad para crear escenarios diferentes para contrastar toda la información posible. La idea entonces es determinar una base de trabajo sobre la que crear diferentes escenarios de red y, a continuación, poder probar diferentes protocolos de routing.

2.4.1. CORE

Con la idea en mente de simplificar este proceso, los simuladores de red surgen como idea. Un simulador reproducirá el comportamiento de los escenarios de red procurando que los dispositivos estén ejecutándose en situaciones lo más reales posibles. Pero existe un problema, y, es justamente el punto a favor que habíamos considerado en un despliegue real de dispositivos, la obtención de unos resultados lo más próximos a la realidad. En un simulador únicamente se podrán testear protocolos de manera conceptual, no con las implementaciones softwares en los propios equipos, perdiendo entonces el concepto realista y fidedigno que se busca.

La solución es encontrar alguna manera de poder crear escenarios de red variados, pero con unos costes y flexibilidad asumibles. Los emuladores de red proporcionan justamente lo que se busca, ya que ofrece la posibilidad de ejecutar software en entornos gráficos concretos, es decir, poder ejecutar implementaciones de protocolos routing reales sobre escenarios similares a la propia realidad.

Existen diversos emuladores de red [16] con los que poder efectuar el trabajo, con la consigna principal de que sea un emulador de código libre y gratuito, que pueda facilitar el futuro desarrollo. Algunos ejemplos son: IMUNES, Shadow y, CORE, emulador a partir del cual trabajamos, ya que reúne todos los requisitos y hay experiencia en el departamento en el que se realiza el proyecto.

Common Open Research Emulator (CORE) [17] es una herramienta de código libre sobre la que poder simular y visualizar escenarios de red, pudiendo

combinar el entorno virtual con los periféricos y puertos de la máquina anfitriona. Dicha herramienta es frecuentemente utilizada para evaluar representaciones en red, estudios de seguridad y, en general, para realizar todo tipo de investigación acerca de entornos de comunicaciones. Dado que CORE se ejecuta en máquinas LINUX, aprovecha todas los beneficios de la virtualización.

En CORE se puede definir una topología de red cualquiera con diferentes equipos, siendo cada uno de ellos una máquina Linux independiente (virtualizada) que puede ejecutar cualquier software que se ejecute en Linux. Dentro de cada elemento, existen unos Servicios que definen, más allá del SO Linux base, algunos otros programas que se ejecutan en el nodo al arrancar el escenario y su configuración. CORE emula los enlaces que conectan los nodos para representar fielmente escenarios de red.

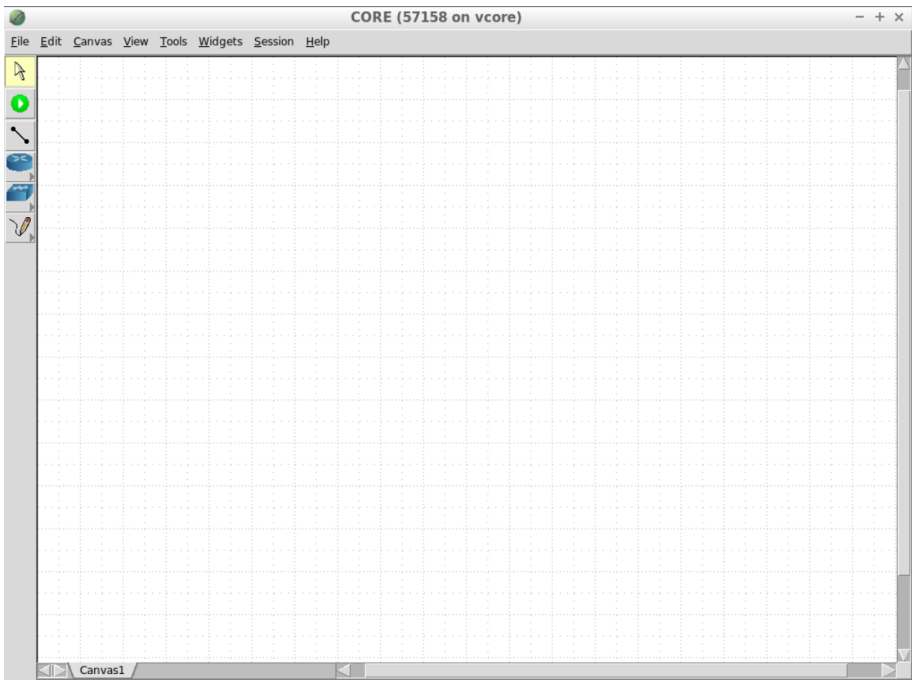


Figura 2.4: Canvas de Core.

Una vez se dispone de la creación de escenarios de red con máquinas Linux, el siguiente reto es encontrar herramientas para poder ejecutar en dichas máquinas protocolos para la transmisión y gestión de tráfico multicast.

2.4.2. IPERF

Iperf [18] es una herramienta de código libre disponible para descarga, ofrecida en paquetes en diversos Sistemas Operativos, entre ellos las máquinas Linux. Se encuentra escrita en lenguaje C y, su principal función es analizar el rendimiento de la red.

Se considera para el trabajo ya que permite transmisión y recepción de tráfico multicast. Entre las posibilidades de Iperf se encuentra el poder medir la latencia, el jitter o variación de retraso y la cantidad de paquetes perdidos en la comunicación.

Una alternativa es crear una aplicación personalizada (como se describe en el apéndice A) para el envío y recepción de tráfico multicast. Se considera como alternativa pero finalmente no se utiliza ya que Iperf ofrece las características necesarias para el proyecto.

2.4.3. XORP

XORP [19] es una herrramienta de software de código libre que implementa protocolos de routing sobre IP. Fue desarrollado en el International Computer Science Institute (ICSI), en Berkeley, California. Está escrito en lenguaje C++ y creado originalmente para *Linux*, pero mantiene soporte para otras plataformas tales como *FreeBSD*,*OpenBSD*, *DragonFlyBSD* y *NetBSD*.

Uno de los beneficios de XORP es que es una plataforma modular que puede ser incluida en múltiples servicios y aplicaciones para realizar estudios de red, ya que soporta protocolos como OSPF, RIP, PIM...

Esta herramienta se instala en Linux mediante paquetes propios. Para su inicialización requiere de una configuración previa, como es el activar ciertos protocolos, por ejemplo, OSPF. Para realizarlo hay dos maneras, todo ello gracias a *xorp_rtrmgr*, que es el *script* que se encarga de la configuración:

- Configurar el router por línea de comandos, escribiendo *./xorp_rtrmgr -h* y, posteriormente, todos aquellos parámetros necesarios para la configuración de cada router.

- Mediante un fichero de configuración, con el cuál se desarrollan en un *script* todos aquellos parámetros para ayudar a la configuración deseada, indicando tanto protocolos como interfaces.

Se dispone de un amplio manual con el que saber usar todas las funciones de los que dispone esta suite e incluso *templates* para facilitar la configuración de los ficheros.

2.4.4. Quagga

Quagga [20] es un software instalable en Linux con el que gestionar diversos protocolos routing en la herramienta como OSPF o RIP. Es una alternativa a XORP, con los mismos objetivos.

Este sistema se basa en el funcionamiento de un demonio llamado *zebra* que se ejecutará a lo largo de los routers dependiendo de la configuración que hayamos seleccionado. Para modificar la configuración, habrá que seleccionar entre los diferentes servicios que provee dicha suite, dando soporte tanto para IPv4 como para IPv6.

La lista de servicios disponibles son:

- RIP y RIPng, para IPv4 e IPv6, respectivamente.
- OSPF, versiones 2 y 3.
- BGPv4 o superiores, incluyendo soporte a IPv6

Existe dentro de Quagga una herramienta para controlar todos los demonios con nombre *vtish*.

2.4.5. PIMB

Pimb [21] es una implementación de un protocolo de routing multicast, PIM.

Está compuesto por dos programas:

- *pimbd*, demonio que ejecuta el protocolo de routing multicast (PIM).
- *pimbc*, programa que se comunica con *pimbd* y permite configurar el funcionamiento del protocolo.

Su funcionamiento se basa en dos pasos, primero un fichero de configuración sobre el que actuará a modo de *socket*, activando PIM en el router designado, con el comando *pimbd*; posteriormente, con *pimbc* se controlará y dará forma a cómo se quiere el funcionamiento de dicho protocolo.

2.4.6. MRD

Multicast Routing Daemon v6 (MRD) [22] es un programa con implementaciones de protocolos de routing multicast con el que probar funcionalidades de red [23].

Algunas características principales útiles:

- Soporte para MLD.
- Soporte para PIM-SM.
- Soporte para interfaces nativas y virtuales

2.4.7. Conclusiones

En esta sección se han presentado diferentes soluciones de movilidad, destacando las características que sobresalen por encima de otras soluciones.

Por otro lado, se introducen los diferentes protocolos de routing, tanto unicast como multicast que proceden para el correcto estudio del proyecto. Dentro de los protocolos de routing unicast que se han estudiado se encuentran RIP y OSPF, mientras que dentro de los protocolos de routing multicast, se destaca a PIM.

Adicionalmente, se presentan implementaciones software de los protocolos anteriormente mencionados, para la consecución de tráfico multicast, como Iperf, PIMB, XORP o MRD6.

Tras la breve presentación de cada elemento, y siguiendo el objetivo del proyecto, se procede en próximos capítulos a analizar la posibilidad de realizar tráfico multicast con la activación de los diferentes protocolos.

Capítulo 3

Marco regulador

Para la idea de la investigación tomamos como referencia la RFC 6224 (Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains) [24], ya que aunque no se aplique en este proyecto, pero es una introducción para comprender el funcionamiento de la gestión de nodos móviles y la recepción de tráfico multicast. En este documento RFC se describen las maneras de configurar nodos en escucha de multicast sin afectar al Dominio PMIP y los estándares de protocolos multicast. Es decir, añade dos elementos como son el LMA como punto de conexión con la red y recepción de tráfico multicast; y el MAG encargado de enviar y recibir todos los mensajes MLD en la subred, gestionando los grupos multicast. A partir de este documento extraemos la información necesaria para poder estudiar los efectos de la movilidad bajo transmisión multicast, es decir, nos ayuda a entender tanto el proceso desde que una fuente puntual emite tráfico multicast y un nuevo MN comienza a recibirlo, hasta los elementos necesarios para hacerlo posible. Algunos ejemplos de información obtenida, pueden ser la necesidad de anuncio de mensajes MLD para descubrir nuevos nodos interesados en grupos multicast o la creación de RP como punto de convergencia del tráfico de la red.

Para la comprensión óptima del anterior documento es necesario estudiar otra RFC, es decir, leer acerca del funcionamiento de Proxy Mobile IP y comprender de mejor manera los elementos y términos que la componen. El documento que aplica es la RFC 5213 (Proxy Mobile IPv6) [6].

Dentro de las dos posibles variantes de PIM, recordemos PIM-SM y PIM-DM, este proyecto sigue un funcionamiento parejo a la primera variante, *Sparse*

Mode. Para su comprensión se aplican ideas según la RFC 2362 (Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification) [25], en la que trata el modo de funcionar de esta variante, con la idea de que la mayor parte de nodos móviles de una subred no requieren la recepción de tráfico multicast. Crean un punto al que se transmitirán todos los paquetes, denominado RP.

Capítulo 4

Desarrollo e implementación

El objetivo del proyecto es estudiar el efecto de la movilidad en tráfico multicast. Primero se necesita un entorno de pruebas sobre el que se puedan efectuar una serie de acciones:

- Crear topologías diversas de red.
- Disponibilidad de encaminamiento de tráfico unicast y multicast.
- Transmitir y recibir tráfico multicast.
- Movilidad de nodos en los escenarios.

Cada acción va a ser analizada individualmente, explicando la manera con la que se ha llegado a conseguir cada una.

4.1. Entorno de pruebas

La elección del escenario de trabajo se basa en los criterios mencionados anteriormente, como son una obtención de unos resultados próximos a la realidad y un coste mínimo. El resultado de investigación entre emuladores nos lleva a la selección de Common Open Research Emulator (CORE) por encima de otras.

Dicha herramienta reúne las características que se buscan para el proyecto y, adicionalmente, el departamento de la Universidad dispone de experiencia de uso previa.

Algunas características principales que provee CORE son:

- Disponer de un laboratorio de red en un entorno emulado.
- Una interfaz gráfica intuitiva.
- Poder utilizar protocolos implementados para un sistema operativo de propósito general, en particular Linux, sin necesidad de desarrollarlos externamente.

4.2. Generación de tráfico

El primer paso es generar tráfico unicast en cada router. Se cominza sobre un escenario básico, con dos nodos host, uno que transmita y otro que reciba tráfico; y un router que sirva de punto de unión.

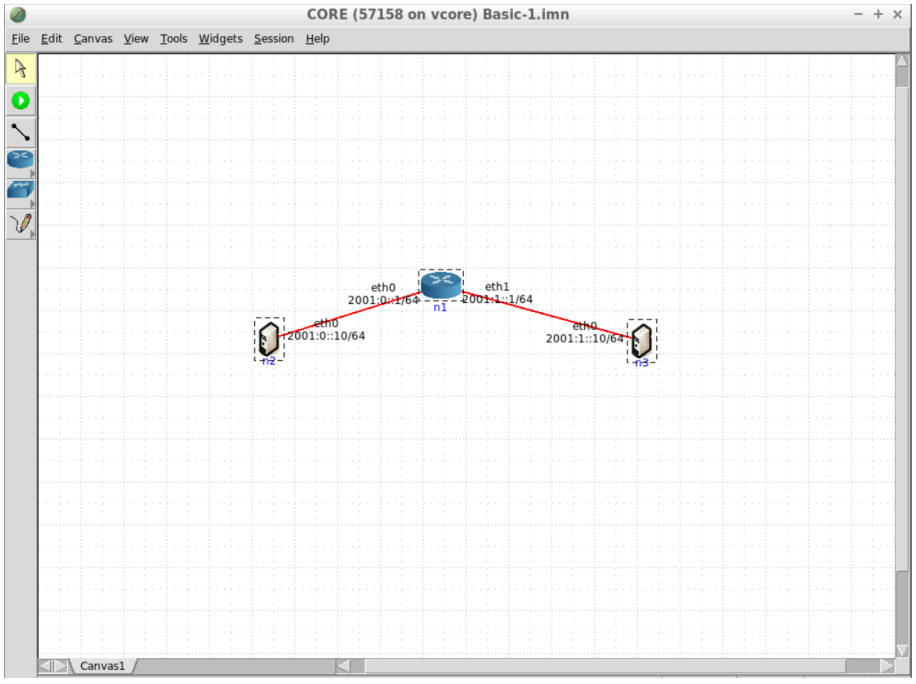


Figura 4.1: Escenario básico de red.

4.2.1. IPERF

En el Estado del Arte (sección 2.4.2) se explica Iperf como la herramienta que se utiliza para la transmisión de tráfico. Los motivos para justificar su uso es

la posibilidad de transmitir tráfico unicast, pero igualmente que incluye soporte a transmisiones multicast, debido a la necesidad más adelante para cumplir el objetivo del proyecto.

La primera decisión acerca de su configuración es decidir si se quiere utilizar Transmission Control Protocol (TCP) o User Datagram Protocol (UDP). Básicamente, TCP usa procesos para comprobar que los paquetes son transmitidos correctamente al receptor mientras que en UDP los paquetes son transmitidos sin ningún tipo de comprobación. La finalidad del proyecto es utilizar tráfico multicast, por lo que se debe elegir UDP, ya que TCP ofrece comunicaciones punto-a-punto y el tráfico multicast es concretamente multipunto.

Para comprobar el correcto funcionamiento se realizan pruebas con tráfico unicast, ya que se necesita primero de un funcionamiento correcto de éste para posteriormente poder probar tráfico multicast. Para realizar las pruebas, los routers necesitan tener algún procolo de routing unicast activado para poder reenviar el tráfico correctamente. En este caso, se activa en los routers los Servicios de Core el protocolo RIPng, en la sección de Quagga.

Para la configuración del nodo receptor, ejecutamos dentro de un sistema final la siguiente línea de comandos, haciendo *bind* a su interfaz de salida para poder recibir los paquetes. En principio, solo se quiere probar que se reciben paquetes correctamente de manera unicast:

```
iperf -s -u -B 2001:0::10 -V -i 1
```

- *-s*. Especifica que va a actuar como servidor, es decir, receptor de tráfico.
- *-u*. Uso del protocolo UDP.
- *-B*. Indica la dirección IPv6 en la que se va a recibir tráfico. Cuando se realicen las transmisiones multicast, ésta será la dirección multicast a la que se subscribe el servidor.
- *-V*. Uso de direcciones IPv6.
- *-i*. Intervalo de tiempo de para mostrar informes de transmisión.

Del lado del cliente, el nodo que transmite tráfico, se ejecuta dentro del sistema final la siguiente línea de comandos, y, al igual que en el servidor, se envían los paquetes a la dirección local de la interfaz del router al que se quiere transmitir los paquetes:

```
iperf -c 2001:0::10 -V -u -T 10 -t 100 -i 1 -b 1000000000
```

- *-c*. Especifica que va a actuar como cliente, es decir, que envía tráfico.
- El siguiente parámetro es la dirección IPv6 a la que envía tráfico el cliente.
- *-V*. Uso de direcciones IPv6.
- *-u*. Uso del protocolo UDP.
- *-T*. Valor del campo de TTL en los paquetes IPv6 que envía el cliente. El valor deberá ser lo suficientemente grande para que los paquetes puedan dar el número de saltos necesario hasta llegar al destino.
- *-t*. Tiempo total durante el que se está transmitiendo.
- *-i*. Intervalo de tiempo de espera para mostrar informes de transmisión.
- *-b*. Cantidad total de datos a transmitir en la comunicación (es decir, la tasa de envío es el valor de -b entre el valor de -t).

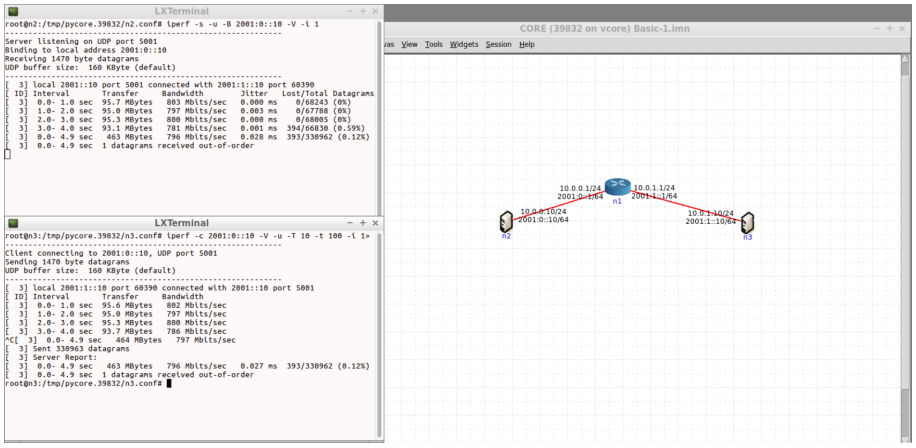


Figura 4.2: Funcionamiento paquete Iperf, cliente y servidor.

Como el funcionamiento de Iperf es totalmente correcto, se envía y recibe tráfico unicast perfectamente, se elige esta herramienta para continuar con el desarrollo.

4.3. Routing Multicast

El siguiente paso del proyecto es poder enviar tráfico multicast. Por el momento, los routers no tienen la configuración necesaria, por lo que se debe activar algún protocolo multicast en los routers del escenario. Se opta por utilizar el protocolo de tráfico multicast PIM y buscar herramientas que lo implementen.

4.3.1. PIMB

Dentro de las herramientas disponibles para Linux del protocolo PIM, se selecciona Pimb, por su facilidad para poder iniciar pruebas de tráfico multicast. En este caso, nuevamente se mantiene un protocolo de routing unicast funcionando, que al igual que en pruebas anteriores, será RIPng.

Para su inicialización, se elige el router que se quiere que actúe como RP, y se inicializa PIM con el comando:

```
pimbd -s /tmp/config.sock
```

Esto hará funcionar el demonio de Pimb dentro del propio router. El parámetro que se introduce a continuación es un fichero que actúa como nombre del socket Unix que los procesos van a utilizar para comunicarse.

Se dispone de PIM activado pero no configurado. Por ello, gracias a otro componente de la herramienta Pimb, se efectúan las configuraciones necesarias para terminar de crear el escenario de routing.

Conectando con *pimbd* para poder configurarlo:

```
pimbc -s /tmp/config.sock link set n1 pim on mld on
```

- *-s*. Enlazamos al fichero que actúa como nombre del socket Unix.

- *link set*. Especifica interfaces o nodo sobre el que se va a realizar la configuración.
- *pim*. Activamos PIM.
- *mld*. Activamos MLD.

Al disponer de conexión en el socket, se comienza a configurar. Primero se indica el router que va a actuar de RendezVous Point (RP), y se fijan las direcciones multicast para las cuáles va a actuar como RP.

```
pimbc -s /tmp/config.sock rpa set 2001:1::1 rpl_jp on
pimbc -s /tmp/config.sock rpa add 2001:1::1 ff1e::/16
```

- *rpa set*. Establece el RP.
- *rpl_jp*. Activa mensajes de unión/desunión del grupo multicast.
- *rpa add*. Añade el prefijo del grupo multicast al RP anteriormente configurado.

Tras tener activado PIM en los routers, se transmite tráfico, pero esta vez utilizando direcciones multicast, no logrando recepción del lado del receptor.

Se estudia el tráfico con *Wireshark* para poder observar en qué punto se quedan los paquetes y averiguar el problema. Primero se captura tráfico en el router desde la interfaz conectada al cliente, el nodo que manda la información. Se observa que los paquetes son correctamente transmitidos.

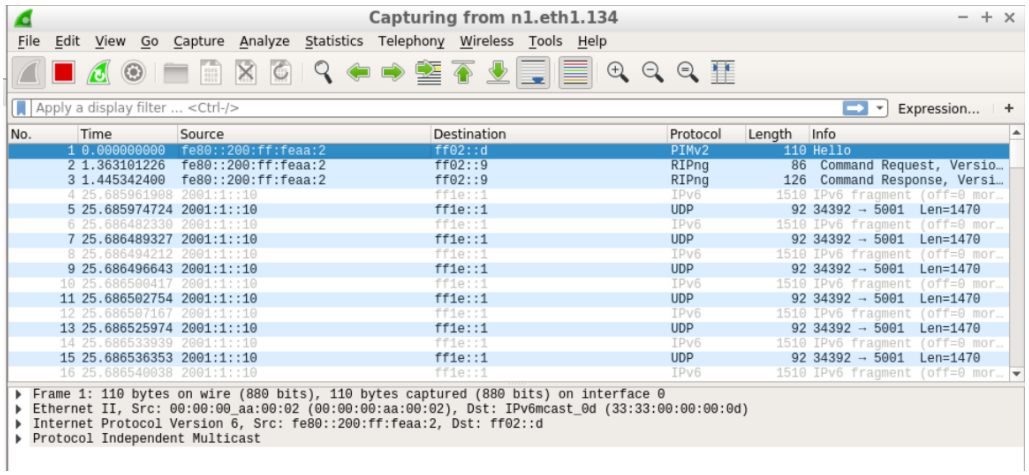


Figura 4.3: Captura de paquetes interfaz cliente.

A continuación, se captura tráfico en la interfaz conectada al nodo que debería recibir paquetes, pero no se observa tráfico. Al menos se puede observar que se intercambian mensajes del protocolo de routing multicast PIM, mostrando que se encuentra activo.

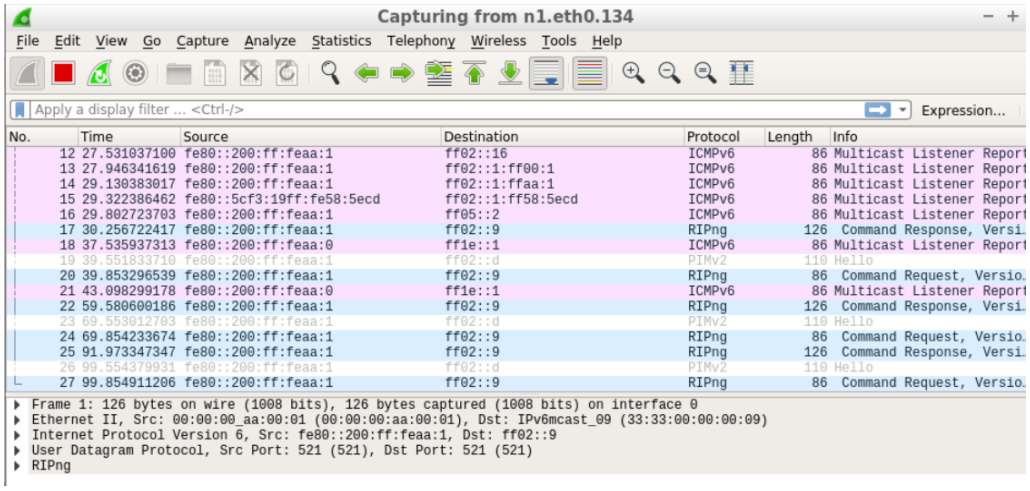


Figura 4.4: Captura de paquetes interfaz servidor con Pimb.

Se crea correctamente el RP según según la siguiente imagen, confirmando que el protocolo PIM actúa correctamente.

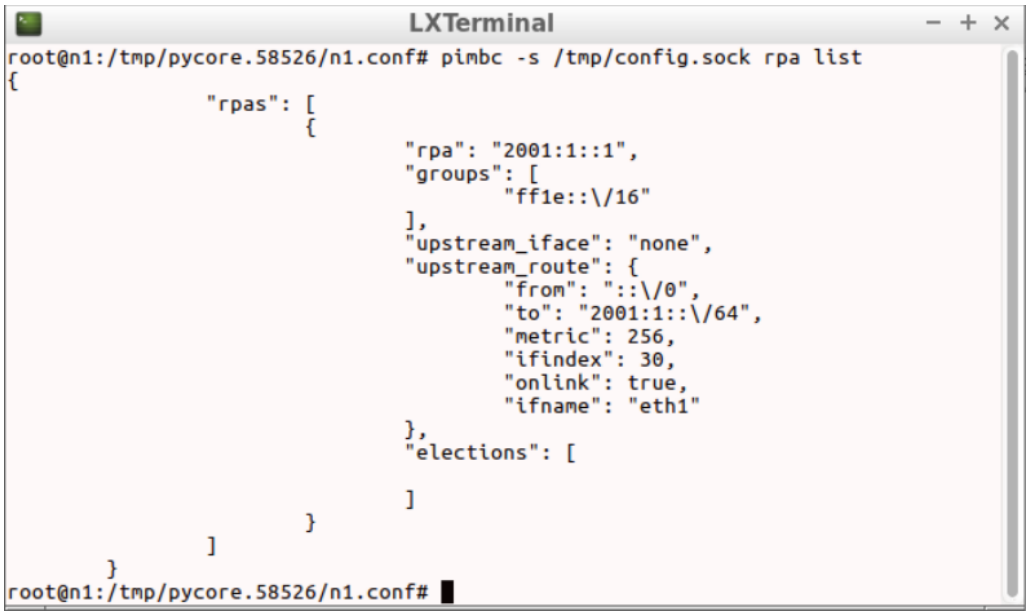


Figura 4.5: RP del escenario con Pimb.

Sin embargo para que los nodos puedan suscribirse a los grupos multicast, necesita el router transmitir mensajes MLD, y en la figura 4.4 no se ve ninguno. Se comprueba si el nodo servidor se suscribe al grupo multicast.

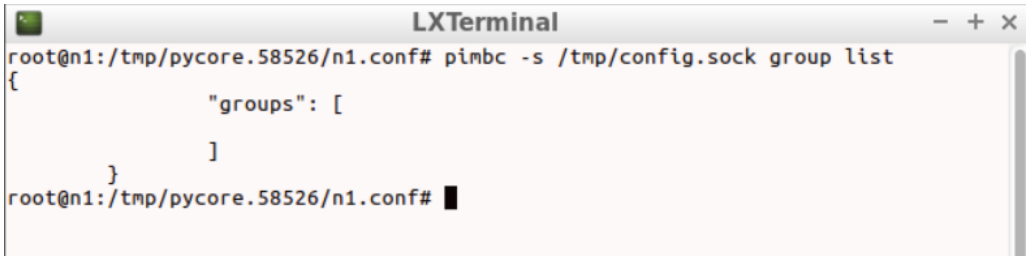


Figura 4.6: Grupo multicast con Pimb.

La imagen 4.6 confirma que el protocolo de routing multicast no funciona correctamente debido a que no hay ningún nodo unido al grupo multicast, impidiendo la recepción del tráfico multicast.

Debido a las limitaciones, no se considera a Pimb para el desarrollo del proyecto ya que no se puede completar uno de los objetivos, transmitir y recibir tráfico multicast.

4.3.2. XORP

Al no poder realizar tráfico multicast con Pimb, se busca una alternativa para continuar con el trabajo, XORP. Esta herramienta dispone de implementaciones de routing tanto unicast como multicast.

Como se dijo en el estado del arte de XORP (descrito en la sección 2.4.3), se puede inicializar con dos métodos: configurando por línea de comandos cada router, con sus parámetros y los protocolos a usar, y, por otra parte, un fichero de configuración que cargará en la inicialización del programa.

Se toma la opción de cargar el fichero de inicialización, ya que al tener varios escenarios de red con varios routers cada uno, la escalabilidad se verá mejorada notablemente respecto a la otra opción.

Se realizan las pruebas con el protocolo de routing multicast PIM activado en los routers. Se muestran las partes esenciales de lo que sería el fichero de configuración inicial, para un determinado router.

- Configuración de las interfaces de los routers, con sus respectivas direcciones:

```
interfaces {
        interface eth0 {
                vif eth0 {
                        address 10.0.0.1 {
                                prefix-length: 24
                        }
                        address 2001::1 {
                                prefix-length: 64
                        }
                        address fe80::200:ff:feaa:1 {
                                prefix-length: 64
                        }
                }
        ...
}
```

- Activación del reenvío unicast a direcciones IPv6:

```
fea {
        unicast-forwarding6 {
                disable:false
        }
}
```

- Se especifican los protocolos a usar, empezando por RIPng, indicando las interfaces sobre las que debe actuar:

```
protocols {
        ripng {
                export: "export-connected"
                interface eth0 {
                        vif eth0 {
                                address fe80::200:ff:feaa
                                   :1 {
                                disable: false
```

```
                                        }
                                }
                        }
                ...
                }
}
```

- Activación de MLD para que el router pueda descubrir receptores multicast:

```
protocols {
        mld {
                ...

                interface eth1 {
                        vif eth1 {
                                disable: false
                        }
                }
                ...
        }
}
```

- Finalmente, configuración de PIM, fijando el RP mediante *bootstrap* (método explicado en la sección 2.2.3):

```
protocols {
        pimsm6 {
                interface eth0 {
                        vif eth0 {
                                dr-priority: 1
                        }
                }
        }
        ...
        bootstrap {
                cand-bsr {
                        scope-zone ff1e::/16 {
```

```
                                        cand-bsr-by-vif-name: "
                                            eth0"
                                }
                        }
                        cand-rp {
                                group-prefix ff1e::/16 {
                                cand-rp-by-vif-name: "eth0"
                                }
                        }
                ...
}
```

Se realiza transmisión de paquetes con Iperf a una dirección unicast, pero no se reciben. Para poder concluir el problema, se investigan las posibilidades de error.

Primero, se realiza un comando *traceroute* entre nodos finales para ver si entienden cómo llegar al destino.

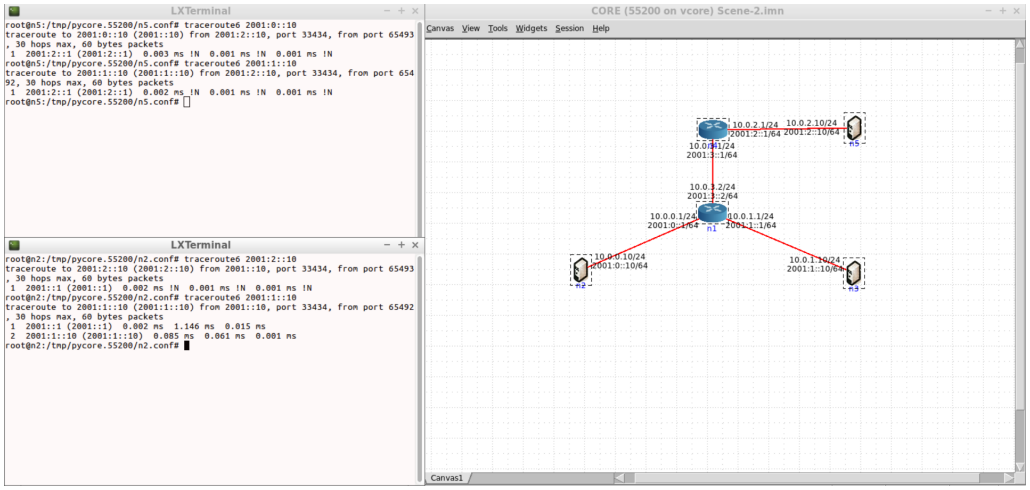


Figura 4.7: *Traceroute* entre nodos. RIP actuando.

No se observan las rutas de más de un salto, por lo que se opta por comprobar las tablas de enrutamiento en la figura 4.8.

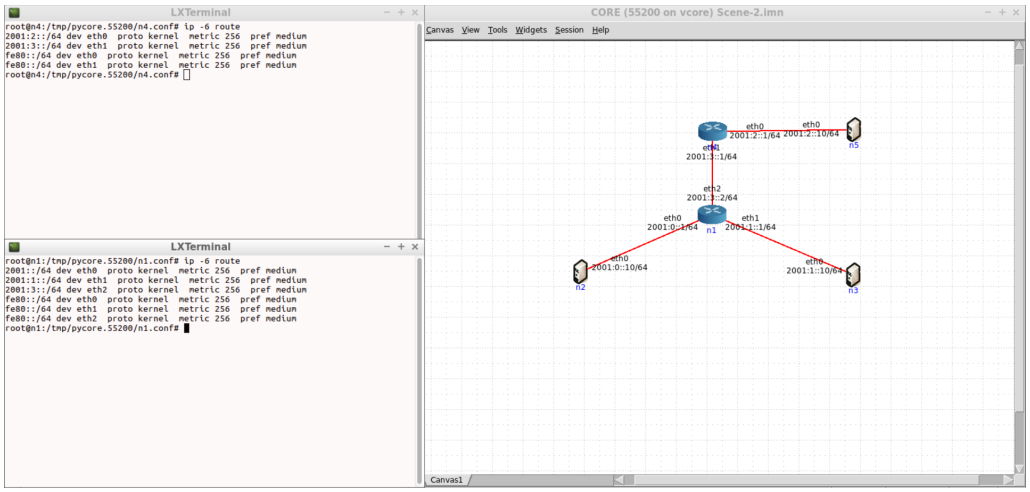


Figura 4.8: Rutas del escenario. RIP funcionando.

El protocolo PIM no funciona correctamente ya que no se crean las rutas para los grupos multicast.

Finalmente, se hace una última comprobación del estado de PIM en todas las interfaces de los routers, para comprobar el RP y poder sacar una conclusión.

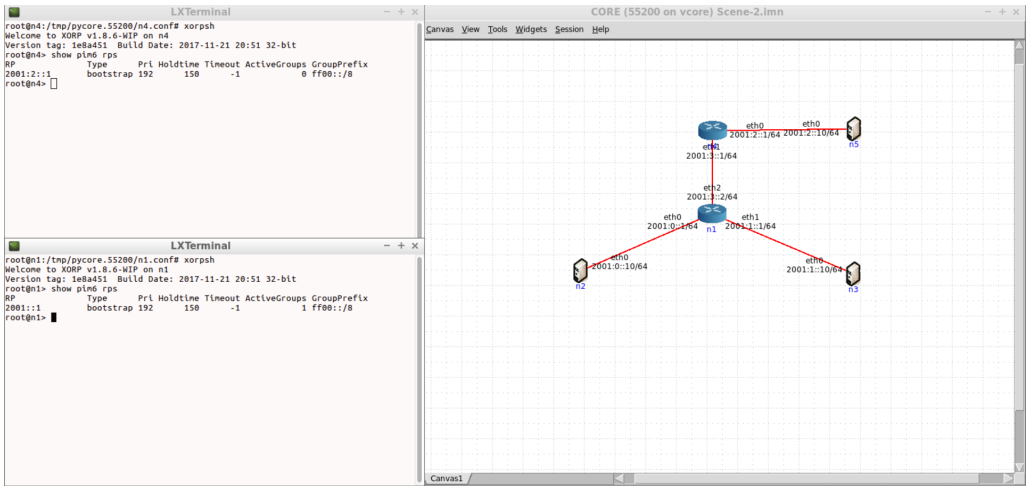


Figura 4.9: RP de la red, RIP en funcionamiento.

Al no estar funcionando correctamente el protocolo unicast RIP (no se puede realizar un *ping* entre nodos que tengan más de un salto entre ellos), es cómo si hubiera dos subredes diferentes, cada una creando su propio RP. La transmisión multicast no es posible con XORP.

Se extrae a los apéndices la misma prueba pero con un protocolo de routing unicast diferente, OSPF. Debido a los problemas de funcionamiento encontrados, se concluye que no es la herramienta adecuada para el proyecto.

4.3.3. MRD

Ante la imposibilidad de transmitir tráfico multicast, se busca una nueva herramienta para realizar este cometido. Se instala MRD6 en el sistema.

Esta herramienta funciona de tal manera que activa un demonio en el interior del router que se ejecute, haciendo que se active en cada router un protocolo de routing multicast, en este caso, PIM, que se encuentra implementado.

Para activar el demonio, hay que ejecutar dentro de cada router el siguiente comando:

```
mrd
```

Activando un protocolo de routing unicast en los Services de cada router (RIPng en este caso), y con PIM activo gracias a MRD, el siguiente paso es intentar transmitir tráfico multicast a varios nodos de la red.

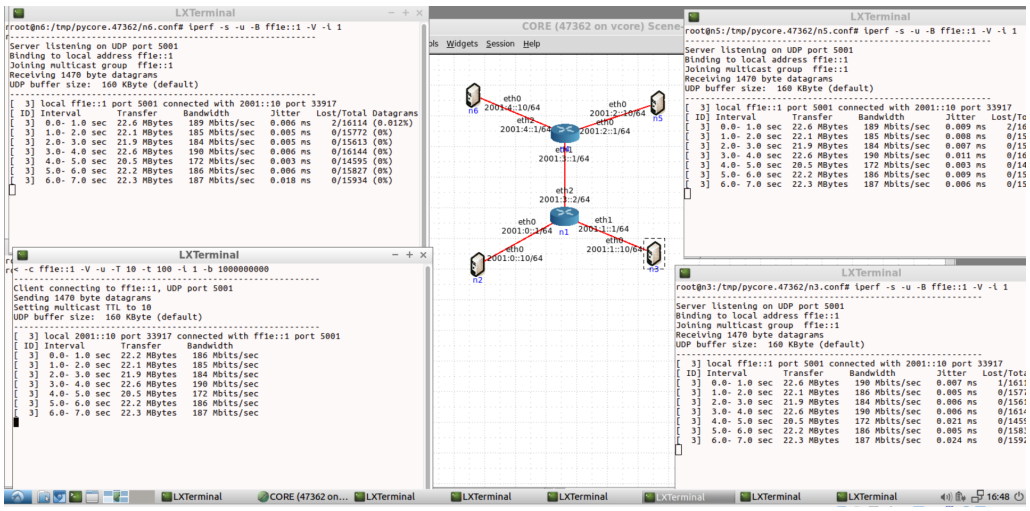


Figura 4.10: Transmisión de tráfico multicast. Demonio de MRD6 activo.

Se confirma la buena transmisión de tráfico multicast, por lo que se obtiene una herramienta que satisface las peticiones de transmisión.

Capítulo 5

Pruebas y resultados

Dado que las pruebas de movilidad se realizan en condiciones inalámbricas, se supone que el MN que va a recibir tráfico no se encuentra previamente adjunto a la red. Por ello, se elimina la dirección IP que tuviera el elemento por defecto y esta dirección será asignada de manera dinámica por los routers de la subred a los que se una. La manera en la que se logra es activando un Servicio propio de los routers, *radvd*, que permite intercambiar mensajes de *Router Advertisement* cuando un nuevo nodo se une a cada router.

Antes de mostrar los resultados obtenidos, conviene puntualizar algunas anotaciones sobre estas pruebas:

- Se realizan un total de 30 pruebas para cada escenario.
- A la finalización de cada prueba, se para al completo la ejecución del escenario, debiendo arrancar éste nuevamente y activando el demonio MRD en cada router para comenzar una nueva prueba.
- Al cambiar nuevamente el MN (en ambos escenarios) como punto de unión a la red al primer router que estuvo unido, la recepción de tráfico se realiza de manera instantánea, sin necesidad de anunciar nuevamente a la red su nueva localización, todo ello transmitiendo y recibiendo en la misma dirección multicast (Se mantienen las rutas).

- En ambos casos, se selecciona como RP en la dirección 2001::1. Esto lo realiza automáticamente el programa que activamos (MRD) para que funcione PIM en cada router.
- Cada prueba se realiza con diferentes direcciones multicast para evitar que pueda afectar una repetición de éstas en los experimentos.
- Para el cálculo posterior de los intervalos de confianza, se establece un nivel de confianza del 95 %.

Las pruebas se realizan sobre dos escenarios diferentes para poder contrastar resultados entre escenarios básicos y más complejos. Primero se comienza con un escenario básico. El nodo n6, según la figura 5.1, será el sistema final que recibe tráfico en condiciones inalámbricas, enviado desde el router n2.

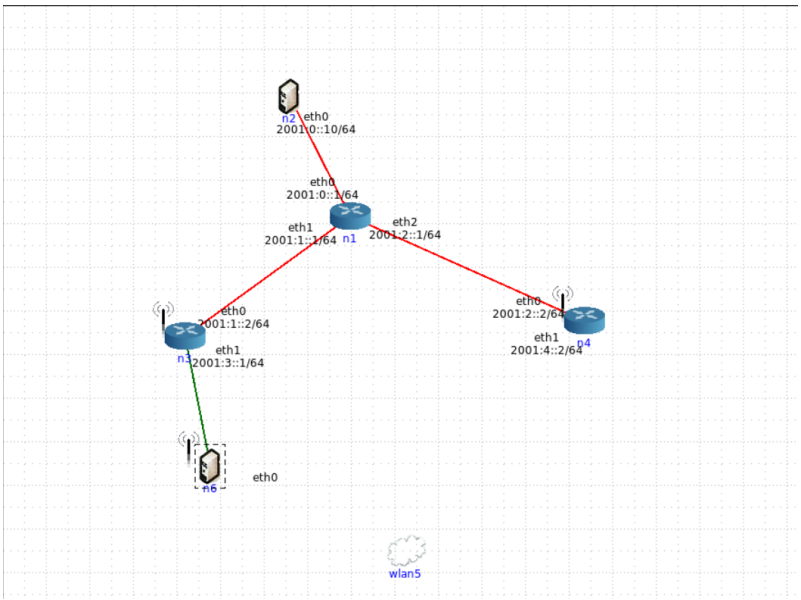


Figura 5.1: Escenario 1 de movilidad.

Mientras se recibe tráfico, se mueve el nodo n6 a la red inalambrica conectada al router n4. Tras unos instantes, el nodo n6 comienza a recibir tráfico de nuevo según se puede ver en la figura 5.2. Se realizan varias pruebas para comprobar plenamente el funcionamiento en movilidad y sacar conclusiones sobre los diferentes parámetros implicados.

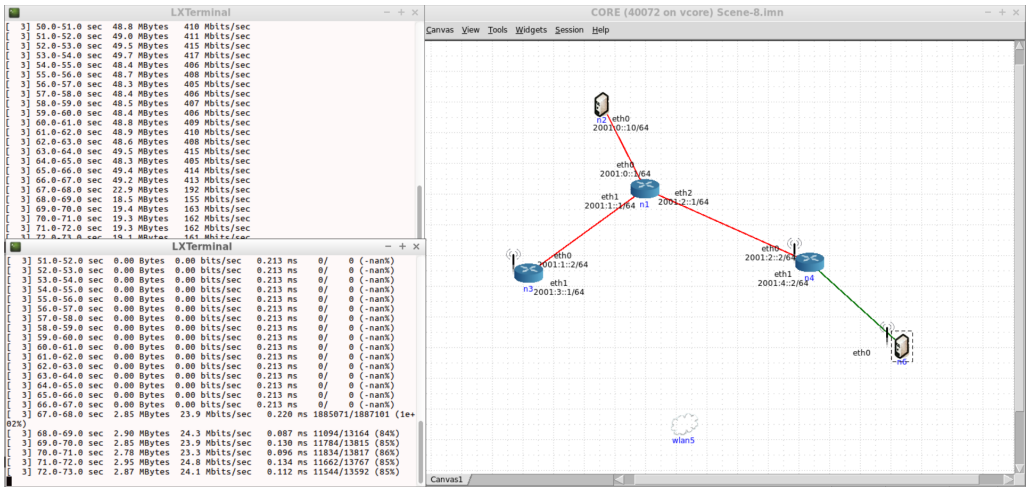


Figura 5.2: Estado del tráfico en movilidad en escenario 1.

Para entender el funcionamiento del proceso, se muestra más adelante una tabla con la diferencia de tiempos sobre los diferentes mensajes capturados en la interfaz del MN. Conviene explicar previamente los datos que se van a medir y el significado de cada palabra de su encabezado.

- *RA*. Corresponde al mensaje de *Router Advertisement* que el router (al que se acaba de unir el MN) transmite para avisar de la aparición del nuevo nodo. Al instante de este mensaje, el MN transmite mensaje de *Neighbor Solicitation* para solicitar la formación del nuevo enlace entre MN y el router.

- *MDNS*. Sus siglas corresponden a *Multicast Domain Name System*, que realiza la función de convertir a direcciones IP los nombres de los host de la red, siendo en este caso, el MN que se une al router. Sirve como medida adicional para comprobar que el enlace está funcionando.

- *MLD*. Se transmiten/reciben dos mensajes MLD, *query* y *report*. El primero, enviado por el router, para encontrar nuevos nodos interesados en tráfico multicast, y el segundo, enviado por el MN para indicar la unión de dicho nodo al grupo multicast.

- *UDP*. Especifica el protocolo de transmisión utilizado durante el proyecto. Los tiempos reflejados en la tabla son desde el momento en el que se comienzan a recibir paquetes.

Tomando como referencia la llegada del RA, se obtienen el resto de tiempos de cada mensaje capturado, mostrando a continuación una tabla comparativa entre estos valores.

	$MDNS - RA$	$UDP - RA$	$UDP - MLD$
Media	41,117872525	80,043874225	1,112514856
Mediana	42,073229770	80,129424206	0,042150483
Intervalo de confianza	±2,032544017	±1,867601197	±0,614829108

Tabla 5.1: Comparativa de tiempos (en segundos) de mensajes en escenario 1.

A continuación, se busca crear un entorno similar a la propia realidad. Por ello, el escenario de red debe contener un número más alto de elementos en la red y con diversas rutas para llegar a un mismo destino. Se añaden fundamentalmente routers ya que son los elementos que afectan principalmente a los protocolos de routing activados, RIP y PIM, pudiendo variar los tiempos por la necesidad de comunicaciones entre ellos.

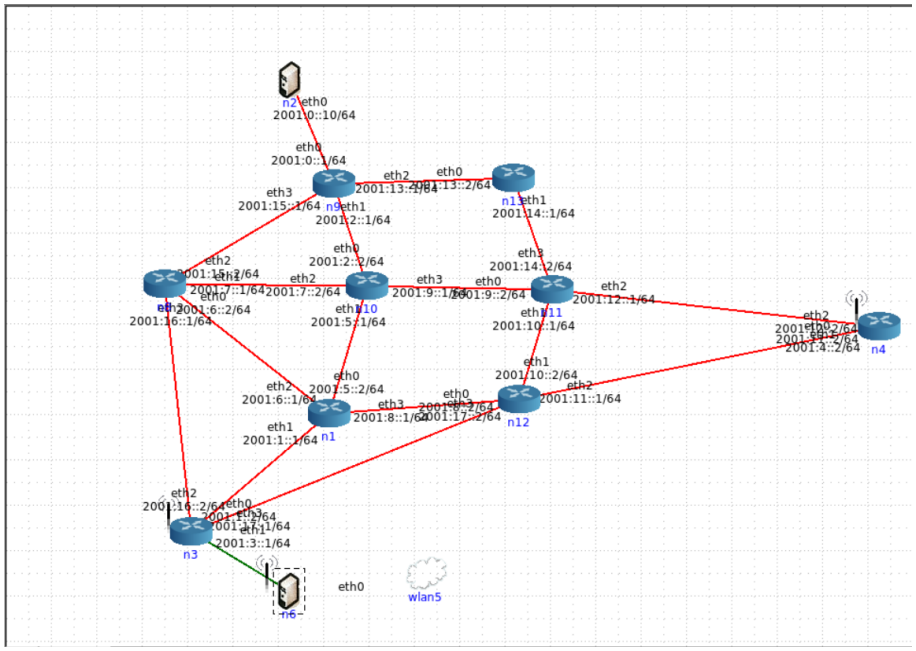


Figura 5.3: Escenario 2 de movilidad.

La recepción de tráfico es totalmente correcta en movilidad a pesar de aumentar el número de elementos en la red.

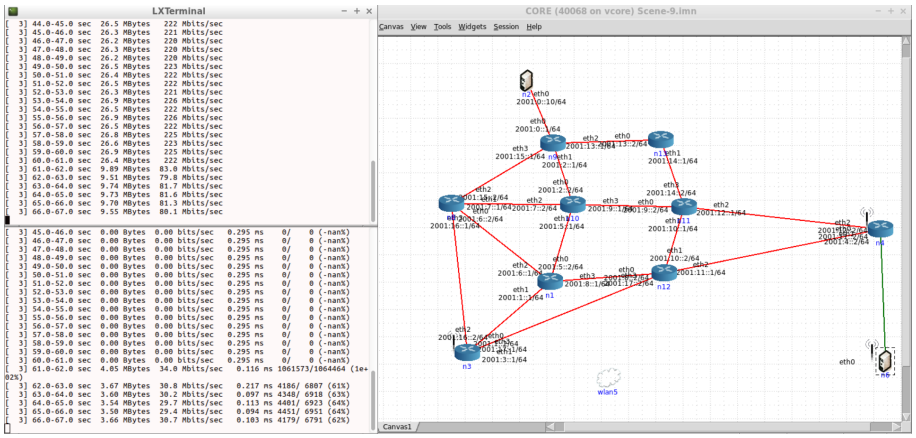


Figura 5.4: Estado del tráfico en movilidad en escenario 2.

Por completar información como en el caso anterior, se toman los tiempos entre los diferentes mensajes que capturamos en la interfaz del MN que conecta con el router y los mostramos en una tabla comparativa.

	$MDNS - RA$	$UDP - RA$	$UDP - MLD$
Media	10,149596122	65,006359091	1,147730001
Mediana	10,713279815	65,405733968	0,042298088
Intervalo de confianza	±3,639494614	±2,718296212	±0,723779269

Tabla 5.2: Comparativa de tiempos (en segundos) de mensajes en escenario 2.

Según las pruebas desarrolladas, la conclusión es que las prestaciones con los protocolos en movilidad son generalmente malas. En ambas pruebas, se confirma por igualdad de tiempos que el tráfico se recibe nada mas enviar un mensaje MLD *Report*. Las diferencias son bastante altas en el momento en el que se reciben mensajes MDNS, que podría afectar a la hora de entender la necesidad de tráfico multicast hacia la interfaz conectada al MN, debiendo estudiar en un futuro la relación de estos mensajes más en profundidad.

Se observa que la diferencia entre la recepción de paquetes UDP y el momento de unión del nodo a otro router (mensaje RA), sufre un retraso de más de 1 minuto en ambos escenarios (siendo incluso 15 segundos superior para el primer escenario). Para entornos reales, estos tiempos serían inadmisibles en términos de calidad de servicio.

En principio, es el envío de MLD *Query* lo que afecta en los retrasos. Según la RFC 2710 (Multicast Listener Discovery for IPv6) [26] estos mensajes son enviados periódicamente, por defecto, cada 125 segundos. Aún así, dentro de la RFC especifica que si un MN comienza a escuchar una dirección multicast en una determinada interfaz, debe enviar inmediatamente un mensaje MLD *Unsolicited Report* para responder su intención de suscribirse inmediatamente al grupo multicast. Por defecto, debería enviar el mensaje al instante el MN cuando cambie a otro router, pero no es el caso. De manera futura se podría estudiar la manera de forzar la transmisión de mensajes MLD cuando los MN cambien de subred, haciendo que se cumpla el funcionamiento establecido según la RFC.

Capítulo 6

Conclusiones y líneas futuras

Finalmente, se ha creado un entorno de pruebas completo que ofrece la posibilidad de estudiar el efecto del tráfico multicast en movilidad. Todo ello ha sido gracias a un emulador de redes y con las herramientas descubiertas para la implementación de los protocolos. Estas herramientas no han funcionado como se esperaba, pero se ha conseguido un conjunto completo de herramientas para Linux que proporcionan routing unicast (RIPng de Quagga, OSPFv3 de XORP), routing multicast (PIM de MRD6), y generación y recepción de tráfico multicast (Iperf).

A pesar de obtener unas pruebas que no han sido óptimas en términos de calidad de servicio, se ha conseguido obtener una base para futuros trabajos e investigaciones.

Existen diversas posibilidades para continuar en base a este estudio, poniendo de ejemplo:

- Crear escenarios todavía más complejos, con un volumen alto de elementos en la red (50-100 elementos).
- Evaluar soluciones específicas para multicast.
- Estudiar casos en los que el MN transmita tráfico en vez de recibirlo.
- Investigar sobre mejorar el tiempo de respuesta al cambiar de enlace (por ejemplo, enviar un Router Solicitation y un MLD *Report* al detectar el cambio de enlace).

- Estudios sobre el tráfico en sí mismo, estudiando vías de mejorar la pérdida de paquetes, o investigar sobre la posible duplicidad de éstos en los intercambios de subred.
- Desarrollar programa de sockets incluyendo tráfico multicast y comparar con actuales herramientas.

Capítulo 7

Summary

Mobile nodes denomination covers many concepts that range from mobile telephones to the growing research of autonomous cars. Thanks to this exponential growth, new ways of improving the transmission bandwidth are being considered, as well as using IPv6 addresses to expand the range of available addresses. With the update to IPv6 addresses it supports full integration of multicast transmissions.

New trends of on-demand services, specially consumed by mobile devices generate an interest in studying the effect of mobility on nodes while maintaining the reception of content. These services transmit (except for specific exceptions) all their content via multicast, so the objective of tis work is studying the multicast mobility traffic.

There are solutions which were developed to deal with unicast traffic mobility, but they also have the possibility to deal with multicast traffic. As an example we have Mobile IP (MIP) and Proxy Mobile IP (PMIP). The former, MIP, faces the problem that IP was developed with the idea that the IP address should be a way to identify devices and locate them in the network. This solution develops the idea of assigning two different addresses to mobile nodes: a permanent home address to identify the node within the Internet, and a care of address that will help node localization.

The other procedure to deal with mobility is PMIP, which instead of using two addresses it uses only one address. It adds two elements in the network, the Local Mobility Anchor (LMA) and the Mobile Access Gateway (MAG), in charge of

managing the multicast traffic that is appropiate for the mobile node, also helping it to move around the subnet.

Given the difficulty of deploying network equipment in real environments, a network emulator, CORE, is used to create networks scenarios that are closer to reality. This tool is generally used to evaluate network features, security studies and conduct research on communication environments. Each element (routers, hubs, wireless networks ...) has internal Services that define programs that run on elements when the scenario starts.

The routers of the scenario must have activated routing protocols to be able to retransmit the traffic they receive. At this point we can not proceed further without introducing the concept of unicast routing protocols, which is mandatory for the correct functioning of multicast routing protocols. The first unicast routing protocol is Routing Information Protocol (RIP). By activating this protocol on routers, the routing table of each router is periodically announced through all of its interfaces. For route management, this protocol uses its own metric based on the number of hops to reach a destination node. The standardized maximum limit is 16 hops, point from which, any network will be considered unachievable. During the project, IPv6 addresses are used, so the updated version, RIPng, is used which extends the funcionalities of RIP first versions, making it compatible with this type of addresses.

The next important unicast protocol is Open Shortest Path First (OSPF). Unlike RIP that sends update messages every 30 seconds to its neighboring nodes, in this case the protocol sends the update messages when it detects changes in the routing table of the routers, sending only the parts that have changed, not the full table. To measure links in this protocol, a metric set in costs is used. That is, an operator can assign different costs depending on the situation of links, being able to assign a lower cost to WLAN links in comparison to a radio-link. In some large, such as corporate networks, this protocol has replaced RIP because of its scalability.

In the case of multicast routing protocols, Protocol Independent Multicast (PIM) is used for its own advantages within its architecture, such as maintaining the traditional multicast group joining model initiated by the receiver and even supporting both shared trees and distribution trees on the shortest path, with its variants PIM-Sparse Mode (PIM-SM) and PIM-Dense Mode (PIM-DM). In PIM-SM is un-

derstood that most subnets are not interested in receiving multicast packets, being the recipients themselves who send explicit messages about the interest of receiving multicast traffic, forming together with the rest of routers and receivers, shared distribution trees. In this variant, a traffic convergence point called RendezVouz Point (RP) is used like a point which receives all the information encapsulated in an unicast manner. For the RP designation, there are two methods: the first one, by static configuration, assigning a fixed IP address to the interface that you want to use as RP, and, the second method, used in this project, selects the Bootstrap Router (BR), which is assigned by the own network.

The second of the variants is PIM-DM, which, unlike PIM-SM, assumes that most receivers require multicast traffic. In this variant, RPs are not used, since source-based trees forms the protocol. It has an efficient behavior when most receivers are interested in multicast data, but not scalable in environments where there are few nodes interested. When PIM-DM is active, it floods all data to the network, forwarding each router traffic through all its interfaces until a request for ending the traffic.

The protocols explained use internal techniques when they are activated. We will explain two of them since they are the most important techniques for multicast and mobility. The first technique is Multicast Listener Discovery (MLD), embedded within ICMPv6. With this technique, subnet's routers discover the final systems interested in subscribing to multicast groups. The other technique is Reverse Path Forwarding (RPF), used by the previous protocols to avoid loops in network. This causes a router to accept a packet from a source in a specific interface, only if that interface is the one that would be used to forward the traffic to reach the mentioned source.

Once the protocols have been theoretically explained, it is interesting to show which is the tool used to transmit traffic and the frameworks that implement the different protocols and allow us to later create the different tests.

At first, the idea is to create a sockets application, with a client program at transmission side and a server program at the receiver side. The program works correctly in a unicast way, but when developing the multicast traffic part, we discover another tool (Iperf) that enables multicast transmission and offers different transmission statistics. The socket application is left as a useful way to learn the

different protocols and try in a future an expansion of the program with the addition of multicast traffic.

Iperf fulfills our requirements in terms of transmission modes (unicast and multicast), and at the time is able to generate statistics regarding of transmission parameters, such as throughpout or number of lost packets. This tool is activated by command line in the final systems of our network scenario and it follows the same scheme as we mentioned before in the socket program, we have a client and a server. In this case, unicast transmission is totally effective, being neccesary to enable a multicast routing protocol in the routers. We need a framework that implements some of the multicast routing protocol.

Various tools are studied in order to get multicast traffic. The first one is Pimb, a tool with PIM implementation. This includes two programs that can be executed from command line: *pimbd*, a daemon in charge of executing PIM protocol, and *pimbc*, a program that communicates with *pimbd* and allows protocol operation configuration. Its operation is based on two steps, first we enable the daemon with *pimbd* and with *pimbc* we assign the point that we want to act as RP as well as the multicast prefix that will be used to form the multicast group within the subnet.

The second alternative is XORP, a tool that implements different routing protocols and specifically among them, those previously explained, RIP, OSPF and PIM. For the initialization, it can be done in different ways, all within each router: the first one is to configure XORP by command line, with all those parameters necessary to activate the different protocols on the interfaces; and second, through a configuration file, in which all the necessary protocols and interfaces are developed in a script, each one being individual and specific for each router.

Third and finally, we study MRD, a tool that offers support funcionalities for MLD and for one of PIM variants, PIM-SM. MRD runs by command line on each router and activates PIM inside it.

Now we begin testing multicast traffic using the different frameworks with Iperf tool to generate traffic and with Quagga RIPng (unicast routing protocol) activated in every router. Using the first alternative, PIMB, multicast traffic is not achieved. Thankfully to PIMB own commands, we can see that the Mobile Node (MN) does not join the multicast group. There is not MLD messages captured in the MN's interface, so multicast transmission is not possible by this way.

We test then multicast traffic using XORP. We choose the way to develop the configuration file for each router because it is better alternative in terms of scalability. We can not achieve multicast traffic due to RIPng problems that create diffents RendezVous Point (RP) in the same subnet. We also try with another routing unicast protocol (OSPF), and in this case we do not have the RP problems but routers do not forward traffic between them. XORP is not an alternative for us.

Finally, like the others alternatives, we activate the unicast protocol RIPng and after activating the daemon MRD in the routers, we begin the data transmission with Iperf tool. The multicast reception is correctly received and end systems susbcribe to the multicast group. It confirms MRD as a valid alternative for the activation of multicast routing protocol.

Now that we have the possibility to transmit and receive data in a multicast way, it is neccesary to create wireless scenarios to get the chance to test the multicast mobility traffic. Therefore, we use CORE which includes the possibility of creating wireless networks and check funcionalities. We create two different scenarios in order to have different approaches to the tests.

We perform several test capturing traffic in the MN interface connected to the routers. We obtain very bad results, due to delays transmissions in mobility (when we move the MN to another subnet). Delays are up to one minute. Frequency MLD messages are the main problem in the performance decrease.

It is possible to continue the mobility research in different ways. Thanks to the experience during this project we can show some of them.

- Evaluate features using specific solutions for multicast traffic
- Improve MLD message response times.
- Study cases in which the MN sends traffic instead of receiving it.

Apéndice A

Programación Sockets

La primera idea sobre cómo generar tráfico unicast es crear una aplicación de sockets de comunicaciones [27]. Realmente, esta idea consiste en dos aplicaciones, una para cliente y otra para servidor. Se desarrolla en lenguaje C.

El servidor es un nodo designado que estará escuchando constantemente todos los paquetes transmitidos a una dirección. Desde el otro lado, el cliente, será quién se encargue de transmitir de manera continuada para que el servidor pueda recibir los datos.

Para que los routers del escenario sepan como reenrutar el tráfico de la red, es necesario que se active dentro de los Servicios propios de cada router un protocolo de routing unicast, siendo para este caso RIPng.

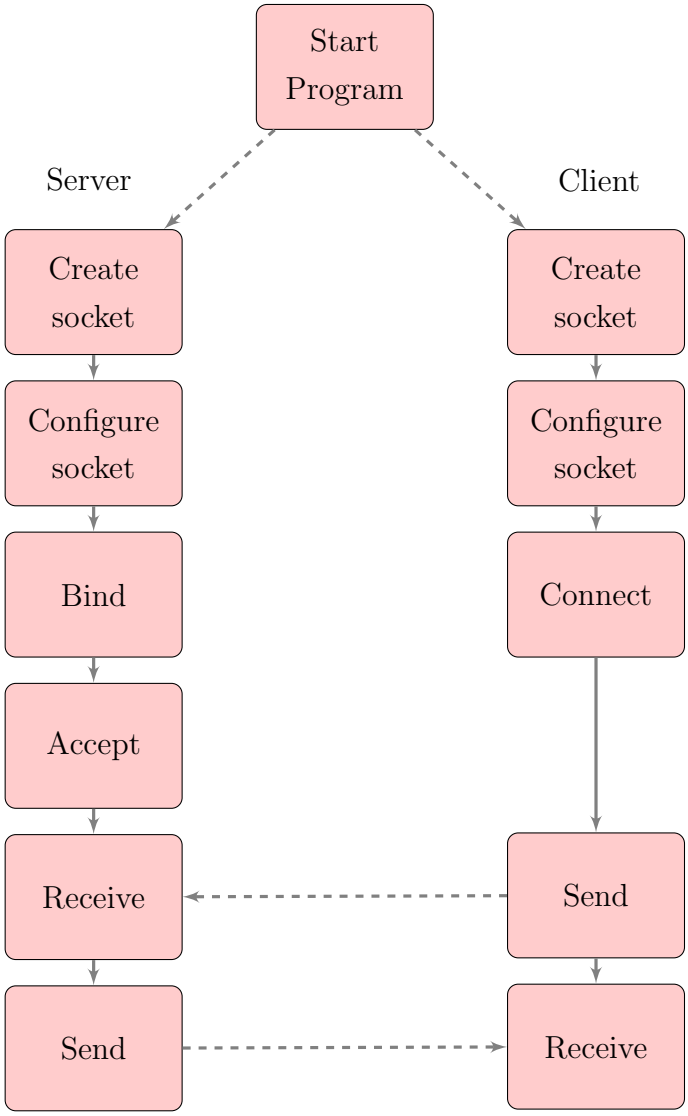


Figura A.1: Estructura de sockets [28].

Una vez vistas las partes fundamentales, lo siguiente es ver el código desarrollado. Primero comenzando con el servidor, situado a la izquierda de la figura A.1. Será el encargado de recibir los paquetes.

- Creación del socket y comprobación de correcta creación. Para la configuración del socket se especifican direcciones IPv6 y transmisión de datagramas.

```
socket_fd = socket(AF_INET6, SOCK_DGRAM, 0);
if (socket_fd < 0){
error("ERROR: Not possible to create a socket");
```

```
        }
```

- El siguiente paso es la configuración del socket. El número de puerto se puede obtener fijando directamente un valor o a través de línea de comandos. En este caso, se fija directamente un puerto definido previamente (puerto 3344 para las pruebas). El sistema lo interpreta como una cadena de caracteres, por lo que se convierte a un entero para que el sistema funcione correctamente. Las siguientes líneas son la configuración del socket, siendo respectivamente, el uso de direcciones IPv6 y, por último, el número de puerto obtenido anteriormente.

```
        //port_number = atoi(argv[1]);
        port_number = atoi(3344);
        printf("Port number: %d",port_number);

        serv_addr.sin6_flowinfo = 0;
        serv_addr.sin6_family = AF_INET6;
        serv_addr.sin6_port = htons(port_number);
```

- Dirección a la que se quiere transmitir tráfico.

```
        host_traffic = gethostbyname2("2001:1::10",
           AF_INET6);
        memmove((char *) &serv_addr.sin6_addr.s6_addr, (
           char *) host_traffic->h_addr, host_traffic->
           h_length);
```

- Se realiza bind a la dirección IP multicast según la configuración del socket. Si el bind se realiza correctamente, se pone al servidor en escucha a la espera de conexiones.

```
        if (bind(socket_fd, (struct sockaddr *) &
           serv_addr,                    sizeof(
           serv_addr)) < 0){
        error("ERROR: Not possible to bind the socket");
        }
        else{
```

```
        printf("\nSocket bound correctly");
        }

        listen(socket_fd, 5);
        printf("\nListening for connections...");
```

- Enlazado al socket y con la dirección IP que va a utilizar el cliente para transmitir (en este caso, la dirección IP del nodo servidor), el buffer es vaciado para almacenar posteriormente los datos recibidos.

```
        inet_ntop(AF_INET6, &(serv_addr.sin6_addr),
           server_addr_ipv6, 100);
        printf("\nConnection from client with IPv6
           address: %s\n",client_addr_ipv6);

        memset(buffer,0, 256);

        client_len = recv(nsocket_fd, buffer, 255, 0);
        if (client_len < 0){
        error("ERROR reading from socket");
        }

        printf("Message from client: %s\n", buffer);
```

- Para evitar posibles fugas de memoria se cierran los socket.

```
        close(socket_fd);
        close(newsocket_fd);
```

Desarrollado el servidor, es necesario enfocar el código del cliente. El cliente es similar a lo realizado en el servidor pero con ciertas modificaciones específicas.

- Se comienza con el número de puerto, que al igual que antes, se da valor previamente y se convierte a binario para que la máquina pueda interpretarlo correctamente. Posteriormente, se crea el socket para direcciones IPv6 y transmisión de datos en forma de datagramas.

```
//port_number = atoi(argv[2]);
port_number = 3344;
printf("Port number: %d",port_number);

printf("\nIPv6 TCP Client Started...\n");

socket_fd = socket(AF_INET6, SOCK_DGRAM, 0);
if (socket_fd < 0){
error("ERROR opening socket");
}
else{
printf("\nSocket opened correctly");
}
```

- Al igual que el puerto, la dirección puede darse a través de línea de comandos, pero se fija en el propio código. Se realiza comprobación de errores para localizar el host. Se configura el socket, fijando direcciones IPv6 y el puerto obtenido anteriormente.

```
//server = gethostbyname2(argv[1],AF_INET6);
server = gethostbyname2("2001:1::10",AF_INET6);
if (server == NULL) {
fprintf(stderr, "ERROR: Not possible to locate
   the host\n");
exit(0);
}

memset((char *) &serv_addr, 0, sizeof(serv_addr))
   ;

serv_addr.sin6_flowinfo = 0;
serv_addr.sin6_family = AF_INET6;
memmove((char *) &serv_addr.sin6_addr.s6_addr, (
   char *) server->h_addr, server->h_length);
serv_addr.sin6_port = htons(port_number);
```

- El socket se conecta según la dirección del servidor.

```
if (connect(socket_fd, (struct sockaddr *) &
   serv_addr,  sizeof(serv_addr)) < 0){
error("ERROR: Not possible to connect to the
   server");
}
else{
printf("\nSocket connected");
}
```

- Para transmitir mensajes por línea de comandos.

```
printf("\nPlease enter the message: ");

memset(buffer, 0, 256);
fgets(buffer,255,stdin);

char_len = write(socket_fd,buffer,strlen(buffer))
   ;
if (char_len < 0) {
error("ERROR: Not possible to write the message")
   ;
}
```

- El buffer se vacía para guardar posibles datos que se reciban.

```
memset(buffer, 0, 256);

char_len = recv(socket_fd, buffer, 255, 0);
if (char_len < 0){
error("ERROR reading from socket");
}

printf("Message from server: %s\n", buffer);
```

- Por último, como con el servidor, se cierra el socket para evitar problemas de fugas de memoria.

```
close(socket_fd);
```

Se ejecutan los programas de cliente y servidor en nodos diferentes de la red. Ambos se conectan al mismo puerto y el receptor debe recibir en su misma dirección IPv6, en este caso, la dirección unicast del nodo que actúa como receptor.

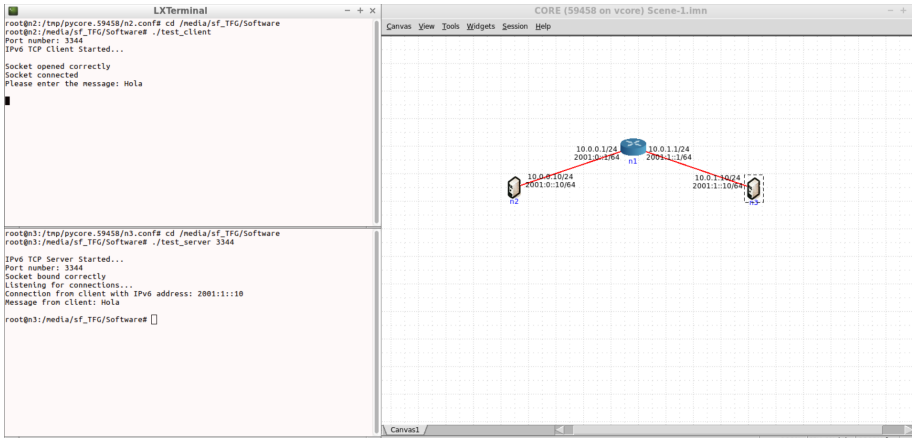


Figura A.2: Ejecución programa de sockets.

La recepción se realiza correctamente, ya que como se ve en la figura A.2 el mensaje escrito del lado del cliente es recibido en la pestaña del servidor.

Se captura tráfico en la interfaz del router conectada al servidor y se observan mensajes UDP, como vemos en la figura A.3, confirmando la correcta transmisión.

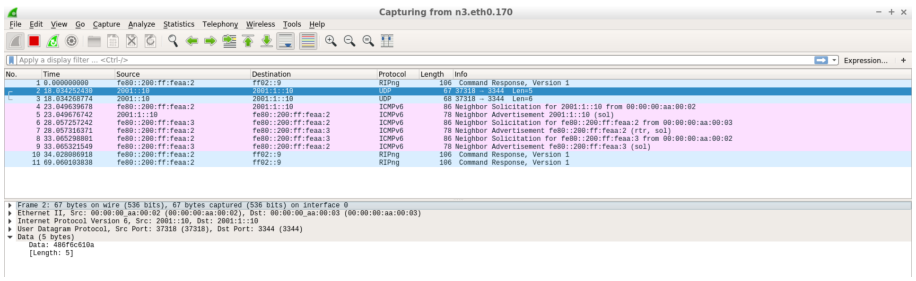


Figura A.3: Captura de tráfico programa de sockets en nodo servidor.

Siendo el objetivo del proyecto es transmitir de manera continuada y a direcciones multicast, se encuentra durante el desarrollo de estos dos programas una herramienta (Iperf, explicada en sección 2.4.2) que puede facilitar las comunicaciones de manera multicast y transmisiones de manera continuada con diferentes ajustes. Se opta por utilizar la nueva herramienta y dejar el programa de sockets como una ayuda al estudio de los fundamentos de las transmisiones tanto unicast como multicast.

Apéndice B

Tráfico multicast XORP con OSPF

Para poder completar el estudio de funcionamiento de XORP, se cambia el protocolo de routing unicast a utilizar, concretamente OSPF

Se cambian ciertas líneas dentro del Servicio de XORP *rtrmgr* de cada router, concretamente las líneas que afectaban al protocolo RIPng por las siguientes:

```
ospf6 0 { /* Instance ID 0 */
router-id: 10.0.2.1
area 0.0.0.0 {
interface eth0 {
vif eth0 {
}
}
interface eth1 {
vif eth1 {
}
}
}
}
```

Se sigue el mismo procedimiento de estudio que el anterior, por lo que comenzamos con una comprobación de las rutas.

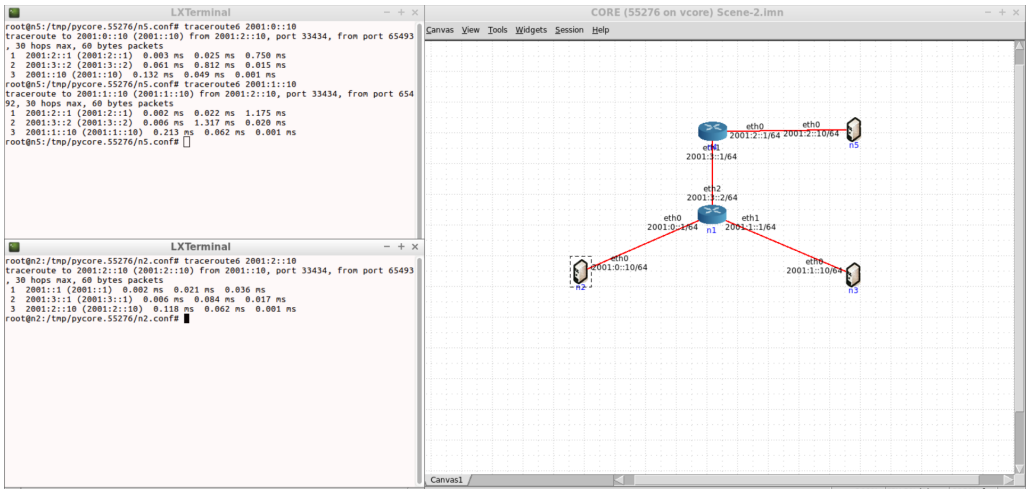


Figura B.1: *Traceroute* entre nodos. OSPF actuando.

Al contrario que con el anterior protocolo, en este caso sí se devuelven correctamente las rutas para destinos de más de un salto. Para terminar de confirmar que el nuevo protocolo está funcionando, se observa la tabla de rutas.

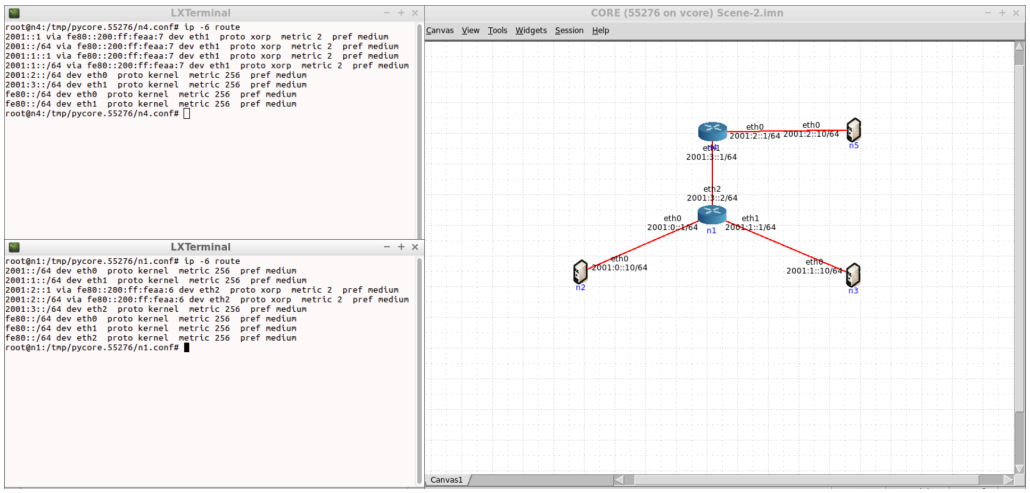


Figura B.2: Rutas del escenario. OSPF funcionando.

Las tablas de enrutamiento son correctamente creadas, con el añadido de que crean correctamente las entradas para los grupos multicast. Se confirma que el protocolo de routing unicast funciona correctamente. Vemos con una última imagen los RP para comprobar la activación de PIM.

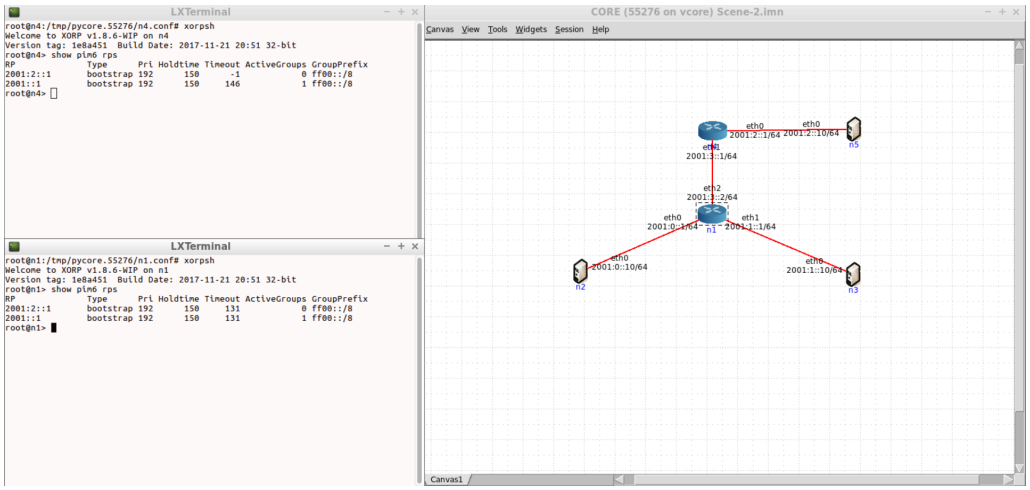


Figura B.3: RP de la red, OSPF en funcionamiento.

El RP es correctamente creado para el escenario, confirmando que el protocolo de routing multicast PIM funciona correctamente en este escenario.

Con todos los protocolos funcionando, únicamente resta por transmitir tráfico multicast.

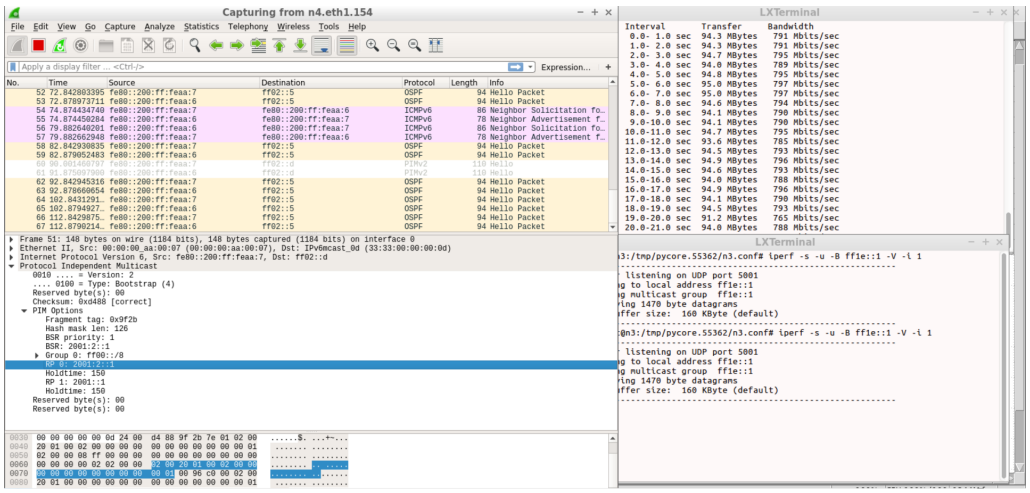


Figura B.4: Captura de tráfico con OSPF funcionando.

Según se ve en la captura, el router no es capaz de retransmitir el tráfico al resto de routers. Se crea correctamente el RP ya que se intercambian correctamente los mensajes de los protocolos OSPF y PIM, pero el tráfico no llega correctamente a su destino.

El protocolo de routing multicast de XORP no reenvía correctamente el tráfico multicast (aunque el estado del protocolo de routing multicast sí parece estar activo y funcionando). Por defecto, el demonio multicast interno de XORP debería activar en el sistema una variable que permita el reenvío de tráfico multicast IPv6. Con XORP no se realiza de esta manera y se intenta tanto introducir un comando en los script de inicialización del escenario, como realizar el cambio manualmente con el escenario inicializado. En ambos casos, no es posible cambiar el valor de la variable, por tanto, no se puede utilizar esta herramienta para ofrecer multicast en los escenarios.

Apéndice C

Entorno socio-económico

El proyecto no busca un impacto inmediato de beneficios económicos. Principalmente se enfoca en la investigación de herramientas y soluciones que permitan obtener conocimientos para una futura explotación económica.

Se logran encontrar maneras de subsanar los problemas de la movilidad en recepción de tráfico multicast, extrayendo ideas de movilidad y multicast de una solución como PMIP. Esto podría ser explotado por operadoras de telecomunicaciones en una segunda fase de investigación en la que se estudie las maneras de aplicarlo a las redes actuales.

Algún ejemplo práctico es una mejora en las aplicaciones IPTV [29]. Llevan años con un buen desarrollo pero a la espera definitiva de un impulso final gracias a innovaciones como PMIP. También puede verse en aplicaciones de vídeo bajo demanda con recepciones en tiempo real del contenido. Los usuarios de los servicios serían los beneficiados ya que podrían disponer de este contenido sin interrupciones.

Más alla de aplicaciones para las telecomunicaciones, hay disciplinas que necesitan soporte de redes óptimas para multicast. Un buen ejemplo es el caso de estudio sobre del impacto de protocolos de routing multicast en sistemas de aprendizaje destinados a la salud [30]. Como se relata en el paper, se requiere de un entorno 3D virtualizado en el que puedan situarse los diferentes departamentos (radiología, maternidad...). El staff compartirá la experiencia a través de dispositivos móviles, siendo manejado el escenario por técnicas de tráfico multicast, ya que se ofrecerá una oferta inmersiva con un menor ancho de banda. Este paper enfatiza

finalmente las necesidades de disponer de unas técnicas efectivas, siendo el ámbito de la salud un campo especialmente crítico a problemas en la red.

Apéndice D

Presupuesto

Se realiza una análisis de los costes aproximados que supone el proyecto. Son separados y desglosados en costes directos (mano de obra y materias primas) y costes indirectos (categorizados directamente como otros costes). Se puede ver en la siguiente tabla el coste final:

Tipo		*Costes*(€)
Costes Directos		
	Ingeniero	10.230
	Ingeniero Senior	1.050
	Hardware	150
Costes Indirectos		
	Otros	2.286
TOTAL		13.716

Tabla D.1: Presupuesto.

A continuación, se explica detalladamente la procedencia de cada coste. Los costes directos engloban al gasto de personal y herramientas utilizadas, entendiendo como gasto de personal al alumno y al tutor del proyecto. Se muestra una planificación de las horas llevadas a cabo para cada tarea en la siguiente figura:

Tipo		*Tiempo(horas)*
Abstracción		
	Inicial	30
	Protocolos	24
	Frameworks	15
Redacción		
	Introducción	15
	Estado del arte	85
	Desarrollo	20
	Pruebas	15
Instalaciones		
	Protocolos	6
	Herramientas	30
Desarrollos		
	Sockets	70
	Pimbc	40
	XORP	30
Otros		
	Experimentos	20
	Solución errores	20
	Correciones	35
	Evaluaciones	10
TOTAL		465

Tabla D.2: Planificación de tareas.

Para un alumno graduado en Ingeniería de Sistemas de Comunicaciones se establece un coste por hora de 22€. Con las horas obtenidas en la planificación, se calcula un coste de 10.230€. Para el tutor (ingeniero senior), se estiman 30 horas, con un coste por hora de 35€, para un total de 1.050€.

A nivel de software, los frameworks han sido de código libre para abaratar costes. Como hardware se ha utilizado un portátil de marca Xiaomi, valorado en 750€. Contando con el coeficiente de amortización anual de productos electrónicos según el Boletín Oficial del Estado para el Impuesto sobre sociedades [31] con un

valor del 20 %, se obtiene un coste de 150€. Para desarrollar la memoria se ha realizado en LaTeX, con un programa de código libre denominado TeXstudio.

Finalmente, los costes indirectos se asignan en una medida proporcional a los costes directos, estableciéndolo en un 20 %. Por tanto, estos costes son de 2.286€.

Acrónimos

AS Autonomous System

BC Binding Cache

BGP Border Gateway Protocol

CN Core Network

CORE Common Open Research Emulator

DHCP Dynamic Host Configuration Protocol

DR Designated Router

HA Home Agent

HNP Home Network Prefix

ICSI International Computer Science Institute

IGP Interior Gateway Protocol

IPTV Internet Protocol Television

LMA Local Mobility Anchor

LMAA LMA Address

LSA Link State Advertisement

MAG Mobile Access Gateway

MH-MN Multihomed Mobile Node

MIP Mobile IP

MLD Multicast Listener Discovery

MN Mobile Node

MN-HL Mobile Node's Home Link

MN-HNP Mobile Node's Home Network Prefix

MN-HoA Mobile Node's Home Address

MN-ID Mobile Node Identifier

MNL-ID Mobile Node Link-Layer Identifier

MRD Multicast Routing Daemon v6

MS Mobility Session

OSPF Open Shortest Path First

OTT Over The Top Technology

PBA Proxy Binding Acknowledgement

PBU Proxy Binding Update

PIM Protocol Independent Multicast

PIM-DM PIM-Dense Mode

PIM-SM PIM-Sparse Mode

PMIP Proxy Mobile IP

Dominio PMIP Dominio PMIP

PP Policy profile

Proxy-CoA Proxy Care-of Address

RA Router Advertisement

RIP Routing Information Protocol

RP RendezVous Point

SPF Shortest Path First

TCP Transmission Control Protocol

TIC Tecnologías de Información y Comunicación

TTL Time to live

UDP User Datagram Protocol

Glosario

LMA Address

Dirección global a la que nos referimos cuando hablamos de la interfaz del LMA y su punto final de transporte del túnel bidireccional establecido entre el LMA y el MAG. A esta dirección son enviados los PBU

Local Mobility Anchor

Es el punto de unión para el nodo móvil y es quién se encarga de gestionar su estado de enlazamiento. Controla todas las rutas de los MN conectados al PMIP domain. Todos los datagramas enviados y recibidos por los MN pasan por el LMA

Mobile Access Gateway

Gestiona la movilidad para el nodo móvil que está conectado a su enlace de acceso. Igualmente, debe rastrear los diferentes movimientos de los nodos móviles a lo largo del dominio y señalizarlo al LMA

Mobile Node

Nos referimos a Mobile Node, a aquellos host IP o routers cuya movilidad es gestionada por la red

Multicast Listener Discovery

Componente de IPv6 utilizado para descubrir receptores multicast en los enlaces.

Dominio PMIP

Define el rango por el cual el cual la gestión de la propia movilidad se realiza mediante el protocolo Proxy Mobile IP. Este dominio incluye igualmente el LMA (Local Mobility Anchors) y los MAG (Mobile Access Gateways) entre los que se puede configurar los sistemas de autorizaciones y actualizaciones tales como los PBU (Proxy Binding Updates) en nombre de los nodos móviles.

Proxy Binding Acknowledgement

Mensaje de respuesta enviado por el LMA en respuesta a un determinado Proxy Binding Update recibido por un Mobile Access Gateway

Proxy Binding Update

Dirección global a la que nos referimos cuando hablamos de la interfaz del LMA y su punto final de transporte del túnel bidireccional establecido entre el LMA y el MAG. A esta dirección son enviados los PBU

Proxy Care-of Address

Dirección global configurada como salida en la interfaz de salida del MAG así como el punto final de transporte del túnel entre el LMA y el MAG

Time To Live

Mecanismo para limitar el número de saltos en una comunicación, limitando la vida de ésta para evitar posibles saturaciones en el servicio.

Bibliografía

[1] "Video will account for an overwhelming majority of internet traffic by 2021." http://www.businessinsider.com/heres-how-much-ip-traffic-will-be-video-by-2021-2017-6. [Fecha de acceso: Diciembre 2017].

[2] D. Smud, C. Wigginton, S. Ninan, K. Ramachandran, and P. Mocer, "Connecting the future of mobility." https://www2.deloitte.com/insights/us/en/focus/future-of-mobility/role-of-telecommunications-in-new-mobility-ecosystem.html. [Fecha de acceso: Octubre 2017].

[3] N. Moro, "The importance of Multicast mechanisms." http://www.teldat.com/blog/the-importance-of-multicast-mechanisms/. [Fecha de acceso: Noviembre 2017].

[4] C. Perkins, "IP Mobility Support for IPv4," RFC 5944, Noviembre 2010.

[5] "What is Mobile IP?." http://searchmobilecomputing.techtarget.com/definition/Mobile-IP. [Fecha de acceso: Junio 2017]., Mayo 2007.

[6] Gundavelli, K. Leung, Cisco, V. Devarapalli, Wichorus, K. Chowdhury, S. Networks, and B. Patil, "Proxy Mobile IPv6," RFC 5213, Agosto 2008.

[7] V. K. Gondi, Q. T. Nguyen-Vuong, and N. Agoulmine, "A New Mobility Solution Based On PMIP Using AAA Mobility Extensions in Heterogeneous Networks.," *Network Operations and Management Symposium Workshops, 2008.*, pp. 39–43, Abril 2008.

[8] "RIP (Routing Information Protocol)." http://searchnetworking.techtarget.com/definition/Routing-Information-Protocol. [Fecha de acceso: Noviembre 2017].

[9] "What is Routing Information Protocol (RIP)?." https://www.metaswitch.com/knowledge-center/reference/what-is-routing-information-protocol-rip. [Fecha de acceso: Noviembre 2017].

[10] "OSPF (Open Shortest Path First)." http://searchenterprisewan.techtarget.com/definition/OSPF. [Fecha de acceso: Noviembre 2017].

[11] "What is Open Shortest Path First (OSPF)?." https://www.metaswitch.com/knowledge-center/reference/what-is-open-shortest-path-first-ospf. [Fecha de acceso: Noviembre 2017].

[12] "Multicast Pim Sparse Mode. Network Lesson." https://networklessons.com/multicast/multicast-pim-sparse-mode/. [Fecha de acceso: Octubre 2017].

[13] "Multicast Pim Dense Mode. Network Lesson." https://networklessons.com/multicast/multicast-pim-dense-mode/. [Fecha de acceso: Octubre 2017].

[14] Cisco, "IPv6 Multicast Listener Discovery Protocol.." https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipmulti_lsm/configuration/xe-3s/imc-lsm-xe-3s-book/ipv6-mcast-mld-xe.html. [Fecha de acceso: Enero 2018].

[15] Wikipedia, "Reverse path forwarding.." https://en.wikipedia.org/wiki/Reverse_path_forwarding. [Fecha de acceso: Enero 2018].

[16] "Open-Source Network Emulators." http://www.brianlinkletter.com/open-source-network-simulators/. [Fecha de acceso: Enero 2018].

[17] U.S. Naval Research Laboratory, "Common Open Research Emulator (CORE)." http://www.nrl.navy.mil/itd/ncs/products/core. [Fecha de acceso: Junio 2017].

[18] Jon Dugan and Seth Elliott and Bruce A. Mah and Jeff Poskanzer and Kaustubh Prabhu, "iPerf - The ultimate speed test tool for TCP, UDP and SCTP." https://iperf.fr/. [Fecha de acceso: Noviembre 2017].

[19] "XORP. Open Source Routing Platform." http://www.xorp.org/. [Fecha de acceso: Noviembre 2017].

[20] "Quagga Routing Suite." http://www.nongnu.org/quagga/. [Fecha de acceso: Noviembre 2017].

[21] "Pimbd - A PIM BIDIR dual-stack implementation." https://github.com/Oryon/pimbd. [Fecha de acceso: Septiembre 2017].

[22] H. Santos, "GitHub MRD6." https://github.com/hugosantos/mrd6. [Fecha de acceso: Diciembre 2017].

[23] Wikipedia, "Multicast Routing Daemon v6." https://en.wikipedia.org/wiki/Multicast_Routing_Daemon_v6. [Fecha de acceso: Enero 2018].

[24] T. Schmidt, H. Hamburg, M. Waehlisch, and S. Krishnanl, "Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains," RFC 6224, Abril 2011.

[25] D. Estrin, USC, D. Farinacci, CISCO, A. Helmy, D. Thaler, UMICH, S. Deering, XEROX, M. Handley, UCL, V. Jacobson, LBL, C. Liu, P. Sharma, and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification," RFC 2362, Junio 1998.

[26] S. Deering, C. Systems, W. Fenner, A. Research, and B. Haberman, "Multicast Listener Discovery for IPv6," RFC 2710, Octubre 1999.

[27] "Simple TCP client server sockets application using IPv6 and IPv6." http://www.electronicsfaq.com/2012/12/simple-tcp-client-server-sockets.html. [Fecha de acceso: Junio 2017].

[28] A. Chugh, "IBM. Accepting connections from both IPv6 and IPv4 clients." https://www.ibm.com/support/knowledgecenter/en/ssw_i5_54/rzab6/xacceptboth.htm. [Fecha de acceso: Junio 2017].

[29] F. F. ALQuayed and S. S. Zaghloul, "Analysis and evaluation of Internet Protocol Television (IPTV).," *e-Technologies and Networks for Development (ICeND), 2014 Third International Conference.*, pp. 162–164, Abril 2014.

[30] A. Zarrad and A. R. Mahlous, "A comprehensive case study of the impact of multicast routing protocols on mobile health care training systems.," *e-Technologies and Networks for Development (ICeND), 2014 Third International Conference.*, pp. 70–74, Abril 2014.

[31] A. E. B. O. del Estado, "Impuesto sobre Sociedades. Boletín Oficial del Estado.." http://www.boe.es/legislacion/codigos/abrir_pdf.php?fich=062_Impuesto_sobre_Sociedades.pdf. [Fecha de acceso: Enero 2018].

[32] Cisco, "IP Multicast Technology Overview." https://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html. [Fecha de acceso: Diciembre 2017].